

TUGAS AKHIR - RG141536

PENGEMBANGAN PROGRAM PENGOLAHAN DATA GPS GLOBAL POSITIONING SOFTWARE

MOHAMMAD HADI KUNNUHA
NRP 3513 100 047

Dosen Pembimbing
Akbar Kurniawan, S.T., M.T.

DEPARTEMEN TEKNIK GEOMATIKA
Fakultas Teknik Sipil dan Perencanaan
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR - RG 141536

PENGEMBANGAN PROGRAM PENGOLAHAN DATA GPS GLOBAL POSITIONING SOFTWARE

MOHAMMAD HADI KUNNUHA
NRP 3513 100 047

Dosen Pembimbing
Akbar Kurniawan, S.T., M.T.

DEPARTEMEN TEKNIK GEOMATIKA
Fakultas Teknik Sipil dan Perencanaan
Institut Teknologi Sepuluh Nopember
Surabaya 2017

“Halaman ini sengaja dikosongkan”



FINAL ASSIGNMENT - RG 141536

**DEVELOPMENT OF GPS DATA PROCESSING
PROGRAM
GLOBAL POSITIONING SOFTWARE**

**MOHAMMAD HADI KUNNUHA
NRP 3513 100 047**

**Supervisor
Akbar Kurniawan, S.T., M.T.**

**DEPARTMENT OF GEOMATICS ENGINEERING
Faculty of Civil Engineering and Planning
Sepuluh Nopember Institute of Technology
Surabaya 2017**

“Halaman ini sengaja dikosongkan”

PENGEMBANGAN PROGRAM PENGOLAHAN DATA GPS GLOBAL POSITIONING SOFTWARE

Nama Mahasiswa : Mohammad Hadi Kunnuha
NRP : 3513 100 047
Jurusan : Teknik Geomatika FTSP – ITS
Dosen Pembimbing : Akbar Kurniawan, S.T., M.T.

Abstrak

Seiring berkembangnya jaman, muncul berbagai teknologi baru yang digunakan untuk memperoleh data lebih banyak dan cepat dalam penentuan posisi suatu objek, seperti perlengkapan global positioning system (GPS). Namun, tidak ada pengukuran yang tepat, setiap pengukuran terdapat kesalahan, nilai asli pengukuran tidak pernah diketahui, dan ukuran pasti kesalahan juga tidak diketahui. Dikarenakan kesalahan pada pengukuran, pengamatan secara berulang pada objek yang sama sering memberikan nilai yang berbeda.

GPS (Global Positining System) merupakan sistem satelit yang digunakan untuk navigasi dan penentuan posisi. Pada dasarnya konsep dasar penentuan posisi dengan GPS adalah reseksi dengan jarak, yaitu dengan pengukuran jarak secara simultan ke beberapa satelit GPS. Meskipun sistem GPS memiliki banyak kelebihan, namun masih terdapat kendala dan keterbatasan dalam pemanfaatannya. Pemrosesan data GPS dan penganalisaan hasil bukan suatu hal yang mudah. Sehingga perlu adanya sebuah program yang dapat digunakan untuk menghitung posisi suatu objek dengan akurat. Salah satu bahasa pemrograman yang sering digunakan adalah C++, sehingga program pengolahan data GPS ini menggunakan bahasa pemrograman tersebut.

Program yang telah dibuat merupakan program awal atau purwarupa untuk pengolahan data GPS. Program ini hanya digunakan untuk penentuan posisi GPS dengan metode absolut. Pada perhitungan dengan menggunakan koreksi troposfer memberikan nilai RMSE yang lebih kecil yaitu $\approx 10 - 20$ m daripada perhitungan tanpa menggunakan koreksi troposfer yang memiliki RMSE sebesar $\approx 20 - 30$ m. Dari tiga data pengukuran pseudorange yang menggunakan sinyal yang berbeda, diketahui bahwa sinyal P2 memberikan hasil yang lebih mendekati nilai koordinat yang sebenarnya dengan nilai RMSE < 15 m. Dan pada dua data pengukuran pada objek yang sama yang diamati di waktu berbeda memberikan hasil yang hampir sama dengan nilai resultan selisihnya adalah 1,190 m.

Kata Kunci : GPS, Pemrograman, Pseudorange, Koreksi Troposfer

DEVELOPMENT OF GPS DATA PROCESSING PROGRAM GLOBAL POSITIONING SOFTWARE

Student Name : Mohammad Hadi Kunnuha
Reg. Number : 3513 100 047
Department : Teknik Geomatika FTSP – ITS
Supervisor : Akbar Kurniawan, S.T., M.T.

Abstract

As the times progressed, new technologies emerge that are used to obtain more and faster data in positioning objects, such as global positioning system (GPS) equipment. However, no precise measurements, any measurements have errors, the original value of the measurement is never known, and the exact size of the error is also unknown. Due to errors in measurements, repeated observations on the same object often give different values.

GPS (Global Positioning System) is a satellite system used for navigation and positioning. Basically the basic concept of GPS positioning is resection with distance, ie by simultaneous measurement of distance to some GPS satellites. Although the GPS system has many advantages, but there are still constraints and limitations in its utilization. GPS data processing and results analysis is not an easy thing. So the need for a program that can be used to calculate the position of an object with accurate. One of the most commonly used programming languages is C ++, so the making of GPS data processing program uses this programming language.

The program that has been created is the initial program or prototype for GPS data processing. This program is only used for GPS positioning with absolute method. In calculations using tropospheric correction it gives a smaller standard deviation

which the RMSE value is $\approx 10 - 20$ m than the calculation without using a tropospheric correction which the RMSE value is $\approx 20 - 30$ m. From the three pseudorange measurement data using different signals, it is known that the P2 signal gives results closer to the true coordinate value which the RMSE value is < 15 m. And on two measurement data on the same object observed at different times gives almost equal results with the resultant of difference value is 1,190 m.

Keywords: *GPS, Programming, Pseudorange, Tropospheric Correction*

**PENGEMBANGAN PROGRAM PENGOLAHAN DATA
GPS
GLOBAL POSITIONING SOFTWARE**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Program Studi S-1 Teknik Geomatika
Fakultas Teknik Sipil dan Perencanaan
Institut Teknologi Sepuluh Nopember

Oleh :

MOHAMMAD HADI KUNNUHA
NRP. 3513 100 047

Disetujui oleh Pembimbing Tugas Akhir

1. Akbar Kurniawan, S.T., M.T.
NIP. 1986 0518 2012 02 0087

(Pembimbing)



SURABAYA, JULI 2017

“Halaman ini sengaja dikosongkan”

KATA PENGANTAR

Alhamdulillah Robbil 'Aalamiin. Puji syukur penulis panjatkan kehadirat Allah SWT, yang telah memberikan nikmat, dan rahmat-Nya berupa keimanan, kemudahan dan pertolongan sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul "Pengembangan Program Pengolahan Data GPS : Global Positioning Software".

Terselesainya Tugas Akhir ini, tidak terlepas dari bantuan dari berbagai pihak. Pada kesempatan yang berbahagia ini penulis ingin menyampaikan terima kasih yang sebesar – besarnya kepada:

1. Orang tua tercinta, yang selama ini dengan tulus ikhlas telah mendidik, memberikan doa dan kasih sayangnya setiap saat. Semoga Allah SWT selalu melimpahkan keselamatan dan kebahagiaan dunia akhirat.
2. Bapak Mokhammad Nur Cahyadi, S.T., M.Sc, Ph.D selaku Ketua Departemen Teknik Geomatika - FTSP ITS yang selalu memberikan wejangan untuk menjadi manusia yang berguna kedepannya.
3. Bapak Yanto Budisusanto, S.T., M.Eng. selaku koordinator Tugas Akhir terima kasih atas semua bantuan dan kesempatan yang telah di berikan.
4. Bapak Akbar Kurniawan, S.T., M.T. selaku dosen pembimbing, atas kebaikan dan dedikasinya dalam memberikan bimbingan hingga dapat terselesaikan Tugas Akhir ini.
5. Seluruh Bapak dan Ibu dosen Teknik Geomatika ITS, yang telah memberikan semangat, harapan, dukungan, ilmu dan bimbingannya.
6. Seluruh teman - teman mahasiswa Teknik Geomatika ITS 2013 yang telah menjadi keluarga seperjuangan yang memberikan arti saling memiliki dan rasa saling membantu sejak awal menjadi mahasiswa hingga akhir pengerjaan tugas akhir ini.

7. Seluruh warga HIMAGE-ITS yang telah memberikan rasa persaudaraan dan nyaman saat pengerjaan tugas akhir.
8. Sahabat – sahabat terbaik yang telah masuk kedalam kehidupan saya yang memberikan dukungan dan semangatnya dan menjadi alasan bagi saya untuk berjuang.
9. Semua pihak yang telah membantu , yang tidak mungkin penulis sebutkan satu per satu

Mudah-mudahan Allah SWT berkenan membalas semua kebaikan, dukungan dan bantuan yang telah diberikan.

Dengan segala keterbatasan, Tugas Akhir ini masih jauh dari sempurna. Oleh karena itu, saran dan kritik yang membangun penulis harapkan untuk pengembangan di masa yang akan datang.

Penulis berharap semoga Tugas akhir ini dapat memberikan manfaat bagi semua pihak dan diterima sebagai sumbangan pemikiran dalam pengembangan ilmu pengetahuan.

Surabaya, Juli 2017

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
ABSTRAK	v
LEMBAR PENGESAHAN.....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
DAFTAR LAMPIRAN	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat Penelitian.....	4
BAB II TINJAUAN PUSTAKA.....	5
2.1 Penentuan Posisi Ekstra-Terestris	5
2.2 Global Positioning System (GPS)	6
2.3 Sistem Koordinat Dalam Geodesi Satelit	7
2.3.1 Sistem Koordinat Referensi CIS.....	7
2.3.2 Sistem Koordinat Referensi <i>CTS</i>	8
2.3.3 Sistem Koordinat Referensi Ellipsod.....	8
2.3.4 Hubungan antara <i>CIS</i> dan <i>CTS</i>	9
2.4 Sistem Koordinat Orbital.....	10
2.5 Sinyal GPS.....	12
2.5.1 Penginformasian Jarak (Kode).....	13
2.5.2 Penginformasian Posisi Satelit.....	15
2.6 Data Pengamatan GPS.....	18
2.7 Penentuan Posisi Dengan GPS	19
2.7.1 Penentuan Posisi Absolut.....	22
2.7.2 Penentuan Posisi Diferensial	23
2.8 Kesalahan Dan Bias.....	24
2.8.1 Kesalahan Ephemeris (Orbit).....	25
2.8.2 Bias Troposfer.....	27

2.9	Persamaan Pengamatan Jarak	28
2.10	Bahasa Pemrograman C.....	31
BAB III METODOLOGI		35
3.1	Peralatan Dan Bahan.....	35
3.1.1	Peralatan.....	35
3.1.2	Data.....	35
3.2	Diagram Alir Metodologi Penelitian	35
3.3	Diagram Alir Pengolahan Data.....	37
BAB IV HASIL DAN PEMBAHASAN		43
4.1	Tampilan Program Pengolahan Data GPS.....	43
4.2	Hasil Pengolahan Data Pengamatan Tanggal 23 November 2015	45
4.3	Hasil Pengolahan Data Pengamatan Tanggal 4 Januari 2016	50
4.4	Hasil Pengolahan Data Pengamatan Tanggal 5 Januari 2016	54
BAB V KESIMPULAN DAN SARAN		59
5.1	Kesimpulan.....	59
5.2	Saran	60
DAFTAR PUSTAKA.....		61
LAMPIRAN		
BIODATA PENULIS		

DAFTAR GAMBAR

Gambar 2. 1 Metode Penentuan Posisi Ekstra-Terestris (Abidin, 2007).....	5
Gambar 2. 2 Hubungan Antara CIS dan CTS (Abidin, 2001).....	9
Gambar 2. 3 Visualisasi Geometrik dari Parameter Orbit GPS (Abidin, 2001)	10
Gambar 2. 4 Sistem Koordinat Orbital (Abidin, 2001)	11
Gambar 2. 5 Informasi pada Sinyal GPS (Abidin, 2007).....	12
Gambar 2. 6 Struktur dan Parameter Dasar Komponen Sinyal GPS (Abidin, 2007).....	13
Gambar 2. 7 Contoh Potongan Struktur Kode pada Sinyal GPS (Abidin, 2007)	14
Gambar 2. 8 Prinsip Penentuan Jarak (Abidin, 2007).....	14
Gambar 2. 9 Prinsip Dasar Penentuan Posisi dengan GPS (Abidin, et al., 2002).....	19
Gambar 2. 10 Prinsip Pengamatan Beberapa Satelit (Abidin, et al., 2002).....	20
Gambar 2. 11 Metode Penentuan Posisi dengan GPS (Abidin, et al., 2002).....	21
Gambar 2. 12 Metode Penentuan Posisi Absolut (Abidin, 2007).....	22
Gambar 2. 13 Metode Penentuan Posisi Diferensial (Abidin, 2007).....	23
Gambar 2. 14 Kesalahan dan Bias GPS (Abidin, 2007).....	25
Gambar 2. 15 Kesalahan Ephemeris (Orbit) (Abidin, 2007).....	26
Gambar 2. 16 Pengamatan Jarak (Ghilani & Wolf, 2006)	29
Gambar 3. 1 Diagram Alir Penelitian.....	36
Gambar 3. 2 Diagram Alir Pengolahan Data.....	38
Gambar 4. 1 Proses Pembacaan Data Navigasi.....	43
Gambar 4. 2 Proses Pembacaan Data Observasi	43
Gambar 4. 3 Penampilan Hasil Pengolahan Data.....	44
Gambar 4. 4 Data Koordinat Tiap Epok.....	44
Gambar 4. 5 Hasil Perhitungan Koordinat Titik Pengamatan	45
Gambar 4. 6 Sebaran Koordinat Tiap Epok pada Sinyal C1	48
Gambar 4. 7 Sebaran Koordinat Tiap Epok pada Sinyal P1	49

Gambar 4. 8 Sebaran Koordinat Tiap Epok pada Sinyal P2.....	49
Gambar 4. 9 Sebaran Koordinat Tiap Epok pada Sinyal C1	52
Gambar 4. 10 Sebaran Koordinat Tiap Epok pada Sinyal P1.....	53
Gambar 4. 11 Sebaran Koordinat Tiap Epok pada Sinyal P2.....	53
Gambar 4. 12 Sebaran Koordinat Tiap Epok pada Sinyal C1	56
Gambar 4. 13 Sebaran Koordinat Tiap Epok pada Sinyal P1.....	57
Gambar 4. 14 Sebaran Koordinat Tiap Epok pada Sinyal P2.....	57

DAFTAR TABEL

Tabel 2. 1 Isi dari <i>Broadcast Ephemeris</i> GPS (Abidin, 2007)	16
Tabel 2. 2 Algoritma Penentuan Koordinat Satelit (Abidin, 2007)	17
Tabel 2. 3 Efek dari Proses Pengurangan Data (Abidin, 2007)...	24
Tabel 2. 4 Nilai Tipikal Kesalahan Orbit GPS (Abidin, 2007) ...	27
Tabel 4. 1 Hasil Perhitungan Tanpa Koreksi Troposfer	46
Tabel 4. 2 Hasil Perhitungan Dengan Koreksi Troposfer	46
Tabel 4. 3 Keakuratan Hasil Koordinat Tanpa Koreksi Troposfer	47
Tabel 4. 4 Keakuratan Hasil Koordinat Dengan Koreksi Troposfer	47
Tabel 4. 5 Hasil Perhitungan Tanpa Koreksi Troposfer	50
Tabel 4. 6 Hasil Perhitungan Dengan Koreksi Troposfer	50
Tabel 4. 7 Keakuratan Hasil Koordinat Tanpa Koreksi Troposfer	51
Tabel 4. 8 Keakuratan Hasil Koordinat Dengan Koreksi Troposfer	51
Tabel 4. 9 Hasil Perhitungan Tanpa Koreksi Troposfer	54
Tabel 4. 10 Hasil Perhitungan Dengan Koreksi Troposfer	54
Tabel 4. 11 Keakuratan Hasil Koordinat Tanpa Koreksi Troposfer	55
Tabel 4. 12 Keakuratan Hasil Koordinat Dengan Koreksi Troposfer	55

“Halaman ini sengaja dikosongkan”

DAFTAR LAMPIRAN

Format Data RINEX Navigasi
Format Data RINEX Observasi
Script Program

“Halaman ini sengaja dikosongkan”

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini manusia hidup di jaman informasi. Dengan adanya dukungan dari teknologi baru yang bermunculan, data dapat diperoleh dengan jumlah yang lebih banyak dari sebelumnya. Seperti pada survei lapangan, peralatan *total station*, perlengkapan *global positioning system* (GPS), kamera metrik digital, dan sistem penginderaan jauh merupakan beberapa peralatan baru yang digunakan untuk memperoleh data dalam jumlah banyak dengan cepat. Sebelum data digunakan baik untuk pekerjaan survei atau pemetaan, harus dilakukan pemrosesan terlebih dahulu. Salah satu aspek yang harus diperhatikan adalah pada semua pengukuran terdapat kesalahan. Dapat dinyatakan bahwa tidak ada pengukuran yang tepat, setiap pengukuran terdapat kesalahan, nilai asli pengukuran tidak pernah diketahui, dan ukuran pasti kesalahan juga tidak diketahui (Ghilani & Wolf, 2006).

Ketelitian data pengamatan bergantung pada besarnya skala alat yang digunakan. Akan tetapi ketelitian juga bergantung pada faktor lain seperti kemampuan dan kondisi alat yang digunakan, kondisi lingkungan yang diamati, dan keterbatasan manusia. Dengan semakin berkembangnya peralatan survei, meningkatnya kondisi lingkungan dan kemampuan pengamat yang meningkat, maka data pengamatan yang diperoleh akan semakin mendekati dengan nilai yang benar, akan tetapi tetap tidak akan sesuai dengan kondisi aslinya. Perbedaan antara nilai pengukuran dan nilai aslinya ini disebut dengan kesalahan (*error*). Kesalahan dalam pengukuran berasal dari tiga sumber, yaitu alat, alam, dan manusia. Kesalahan tersebut dihasilkan dari kondisi fisik yang menyebabkan terjadinya kesalahan secara sistematis dan juga

terjadi secara acak. Sehingga kesalahan dapat diklasifikasikan menjadi kesalahan besar, sistematis, dan acak.

Dikarenakan kesalahan pada pengukuran, pengamatan secara berulang pada objek yang sama sering memberikan nilai yang berbeda. Perbedaan nilai yang sangat kecil pada beberapa pengukuran pada objek yang sama dipercaya dapat memberikan nilai kesalahan yang kecil, sehingga pengukuran tersebut dianggap presisi. Namun pengukuran yang presisi tidak selalu memberikan nilai pengukuran yang mendekati nilai sesungguhnya. Pengukuran yang memiliki nilai mendekati nilai sesungguhnya disebut dengan akurasi. Pengamatan yang dilakukan secara berulang memberikan data pengukuran yang berlebih. Dengan data pengukuran yang berlebih tersebut, dapat diperoleh nilai akhir yang mendekati nilai sesungguhnya dengan menggunakan metode perataan. Perhitungan perataan dilakukan untuk mengetahui adanya kesalahan pada pengamatan dan pertambahan presisi nilai akhir yang dihitung. Berbagai metode berbeda telah diberikan untuk melakukan perataan pada data survei, akan tetapi metode kuadrat terkecil paling sering digunakan karena memberikan nilai kesalahan paling kecil dari metode lain. Metode perataan kuadrat terkecil juga dipergunakan untuk mengetahui kesalahan pada pengukuran GPS dan untuk melakukan perataan pada data jaring GPS.

GPS (*Global Positioning System*) merupakan sistem satelit yang digunakan untuk navigasi dan penentuan posisi yang dimiliki oleh Amerika Serikat. GPS didesain untuk memberikan informasi posisi dan kecepatan tiga dimensi serta informasi waktu. Sistem GPS berjalan secara terus – menerus di seluruh dunia dan tidak terganggu oleh waktu dan cuaca. Sehingga GPS banyak dimanfaatkan terutama dalam bidang penentuan dan perubahan posisi (Abidin, 2007). Sistem GPS memiliki lebih banyak kelebihan dibandingkan sistem dan metode penentuan posisi lainnya, baik dalam segi operasional maupun kualitas posisi yang didapatkan. Tidak hanya sistem

satelit GPS saja yang saat ini digunakan, namun juga terdapat GLONASS milik Rusia dan Galileo milik Eropa. Metode penentuan posisi dengan menggunakan pengukuran atau pengamatan ke objek di angkasa seperti GPS ini disebut metode ekstra terestris. Meskipun sistem GPS memiliki banyak kelebihan, namun masih terdapat kendala dan keterbatasan dalam pemanfaatannya. Pemrosesan data GPS dan penganalisaan hasil bukan suatu hal yang mudah. Dalam melakukan pemrosesan data untuk memperoleh ketelitian posisi yang tinggi, digunakan konsep hitungan perataan kuadrat terkecil, statistika, serta perhitungan geodetik.

1.2 Rumusan Masalah

Adapun permasalahan yang dimunculkan pada penelitian ini adalah bagaimana cara membuat suatu program pengolahan data GPS dengan menggunakan bahasa pemrograman C++.

1.3 Batasan Masalah

Adapun batasan masalah dari penulisan tugas akhir ini adalah sebagai berikut :

- a. Program yang dibuat merupakan program yang digunakan untuk mengolah data GPS.
- b. Program yang dibuat hanya dikembangkan untuk penentuan posisi absolut.
- c. Data yang digunakan merupakan data berformat RINEX.
- d. Pembuatan program menggunakan perangkat lunak Dev-C++.

1.4 Tujuan

Tujuan tugas akhir ini adalah untuk membuat sebuah program pengolahan data GPS hasil pengukuran dengan menggunakan bahasa pemrograman C++.

1.5 Manfaat Penelitian

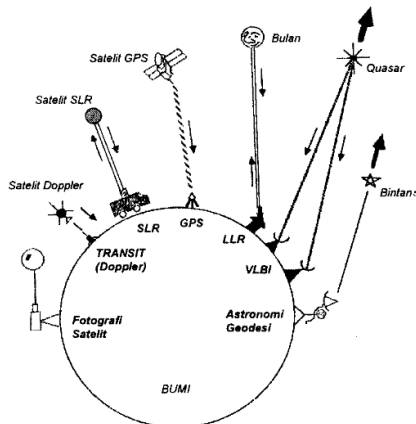
Manfaat dari tugas akhir ini adalah program ini dapat dipergunakan secara langsung untuk pengolahan data hasil pengukuran.

BAB II

TINJAUAN PUSTAKA

2.1 Penentuan Posisi Ekstra-Terestris

Dalam penentuan posisi titik di permukaan bumi terdapat metode terestris dan ekstra-terestris. Metode terestris dilakukan dengan pengukuran dan pengamatan secara langsung di permukaan Bumi. Sedangkan metode ekstra-terestris merupakan metode penentuan posisi dengan melakukan pengukuran atau pengamatan dengan memanfaatkan objek/benda di angkasa, baik yang bersifat alam (bulan, bintang dan quasar) maupun buatan manusia seperti satelit. Ada beberapa metode pada penentuan posisi ekstra-terestris hingga saat ini, seperti astronomi geodesi, fotogrametri satelit, SLR (*Satellite Laser Ranging*), LLR (*Lunar Laser Ranging*), VLBI (*Very Long Baseline Interferometry*), Transit (Doppler), dan GPS (*Global Positioning System*), yang diilustrasikan seperti pada gambar 2.1.



Gambar 2. 1 Metode Penentuan Posisi Ekstra-Terestris (Abidin, 2007)

Saat ini metode penentuan posisi ekstra-terestris yang paling populer dan paling banyak dipergunakan adalah GPS (Abidin, 2007). Namun juga terdapat sistem satelit lain seperti GLONASS milik Rusia dan Galileo milik komunitas Eropa. Saat ini metode fotografi satelit sudah tidak digunakan lagi dan sistem satelit Doppler dan astronomi geodesi sudah jarang dipergunakan. Sedangkan metode SLR, LLR, dan VLBI umumnya digunakan untuk kegiatan ilmiah yang menuntut ketelitian posisi yang sangat tinggi.

2.2 Global Positioning System (GPS)

GPS adalah sistem satelit navigasi dan penentuan posisi menggunakan satelit milik Amerika Serikat. Memiliki nama formal NAVSTAR GPS yang merupakan kependekan dari "*Navigation Satellite Timing and Ranging Global Positioning System*". Sistem ini dapat digunakan oleh banyak orang sekaligus dalam berbagai kondisi cuaca, didesain untuk memberikan posisi dan kecepatan tiga dimensi yang teliti dan juga informasi mengenai waktu dan kontinyu di seluruh dunia (Abidin, 2007).

Satelit GPS beroperasi dengan menggunakan sistem waktunya sendiri. Sistem waktu tersebut didefinisikan oleh jam – jam atom yang berada pada setiap satelit GPS. Semua frekuensi yang dibangkitkan di satelit serta waktu pentransmisian untuk kode C/A, kode P(Y), dan pesan navigasi mengacu pada sistem waktu tersebut. Meskipun begitu perlu dicatat bahwa data yang berada dalam pesan navigasi adalah mengacu ke sistem waktu GPS.

Sistem waktu GPS adalah sistem waktu berskala kontinyu yang didefinisikan oleh jam (atom) utama yang berada di *Master Control Station* (MCS) GPS di Colorado Springs. Sistem waktu GPS bereferensi ke sistem waktu UTC (*Universal Time Coordinated*) yang dikelola oleh USNO (*United States Naval Observatory*). Karena adanya *leap second* pada skala sistem waktu UTC, serta adanya *drift* pada

jam MCS, maka sistem waktu GPS tidak identik dengan sistem waktu UTC. Kedua sistem tersebut dihubungkan dengan sistem waktu IAT (*International Atomic Clock*) dengan persamaan berikut :

$$\text{IAT} = \text{GPS} + 19.^{\circ}000 \quad \text{Persamaan 2.1.}$$

$$\text{IAT} = \text{UTC} + (1.^{\circ}000).n \quad \text{Persamaan 2.2.}$$

Nilai bilangan integer n secara formal diumumkan oleh IERS (*International Earth Rotation Service*). Waktu GPS dinyatakan dengan bilangan minggu serta (jumlah) detik sejak awal minggu pada saat tersebut. Waktu GPS berubah dari 0 pada tengah malam antara Sabtu dan Minggu, sampai 604800 di akhir minggu. Bilangan minggu GPS dimulai pada tanggal 5 Januari 1980 jam 00:00 UTC. Pesan navigasi GPS mengandung parameter – parameter koreksi untuk mengkonversikan sistem waktu satelit ke sistem waktu GPS.

2.3 Sistem Koordinat Dalam Geodesi Satelit

Pada dasarnya ada 3 sistem *referensi* koordinat yang banyak digunakan dalam bidang Geodesi Satelit (Abidin, 2001), yaitu:

1. *CIS (Conventional Inertial System)*
2. *CTS (Conventional Terrestrial System)*
3. Sistem Ellipsoid

Sistem *CIS* umumnya digunakan untuk mendefinisikan posisi dan pergerakan satelit, sedangkan sistem-sistem *CTS* dan ellipsoid untuk mendefinisikan posisi dan pergerakan titik di permukaan bumi.

2.3.1 Sistem Koordinat Referensi CIS

CIS merupakan sistem koordinat referensi yang terikat langit, untuk mendeskripsikan posisi dan pergerakan satelit. Sistem ini tidak berotasi terhadap bumi, tetapi ikut berevolusi bersama bumi mengelilingi matahari. Karakteristik dari sistem ini adalah :

- a. Titik nol sistem koordinat adalah pusat bumi dan sumbu-sumbu sistem koordinatnya terikat ke langit.
- b. Sumbu x mengarah ketitik semi pada epok standar J2000.0 dan terletak pada bidang ekuator bumi.
- c. Sumbu z mengarah ke *CEP* pada epok standar J2000.0 posisi adalah posisi bebas di langit dari sumbu rotasi bumi.
- d. Sumbu Y tegak lurus sumbu X dan Z, dan membentuk sistem koordinat tangan kanan.

2.3.2 **Sistem Koordinat Referensi CTS**

CTS digunakan untuk mendeskripsikan posisi dan pergerakan titik dipermukaan bumi. Sistem koordinat ini berotasi dengan bumi dan juga berevolusi bersama bumi mengelilingi matahari. Sistem ini memiliki beberapa karakteristik :

- a. Titik nol sistem koordinat adalah pusat bumi dan sumbu-sumbu sistem koordinatnya terikat ke bumi.
- b. Sumbu X berada pada bidang meridian Greenwich dan terletak pada bidang ekuator bumi.
- c. Sumbu Z mengarah pada *CTP*, yaitu kutub menengah bola langit pengganti *CIO*. *CIO* adalah posisi rata-rata sumbu rotasi bumi dari tahun 1900 sampai 1905.
- d. Sumbu Y tegak lurus sumbu-sumbu X dan Z, dan membentuk sistem koordinat tangan kanan.

2.3.3 **Sistem Koordinat Referensi Ellipsoid**

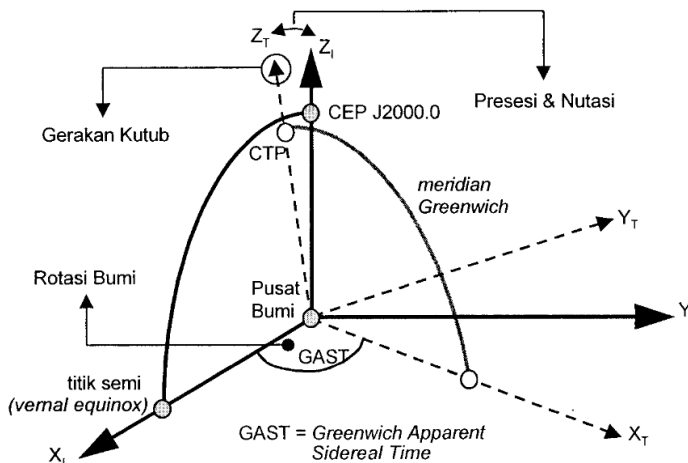
Permukaan bumi dapat didekati secara baik dengan ellipsoid putaran yaitu ellips meridian yang diputar mengelilingi sumbu pendeknya. Oleh sebab itu secara geometrik, koordinat titik-titik dipermukaan bumi juga dapat dinyatakan koordinatnya dalam sistem referensi ellipsoid. Sistem ini berotasi engan bumi dan berevolusi bersama bumi mengeliling matahari. Sistem koordinat

referensi ellipsoid mempunyai karakteristik sebagai berikut :

- Titik nol sistem koordinat adalah pusat ellipsoid.
- Sumbu X berada dalam bidang meridian nol dan terletak pada bidang ekuator ellipsoid.
- Sumbu Z berimpit dengan sumbu pendek ellipsoid.
- Sumbu Y tegak lurus sumbu-sumbu X dan Z, dan membentuk sistem koordinat tangan kanan.

2.3.4 Hubungan antara CIS dan CTS

Sistem-sistem koordinat CTS dan CIS dapat ditransformasikan antar sesamanya dengan menggunakan besaran-besaran presesi, nutasi, gerakan kutub, dan rotasi bumi.



Gambar 2. 2 Hubungan Antara CIS dan CTS (Abidin, 2001)

Seandainya koordinat suatu titik dalam kedua sistem dinyatakan sebagai berikut :

$$X_{CIS} = (X_I', Y_I', Z_I')$$

$$X_{CTS} = (X_T, Y_T, Z_T)$$

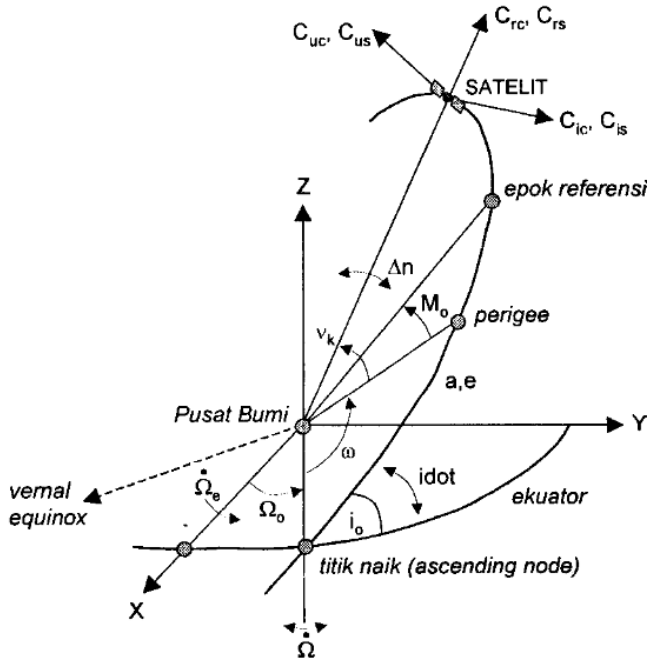
dimana,

M = matriks rotasi untuk gerakan kutub (*polar motion*).

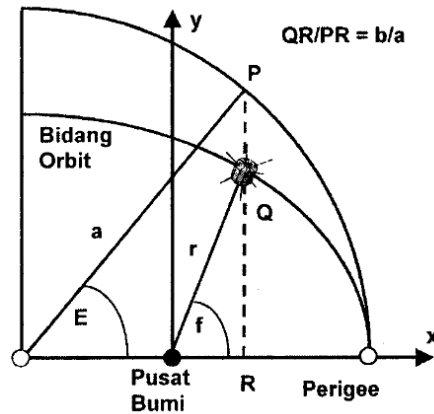
S = matriks rotasi untuk rotasi bumi (*earth rotation*).
 N = matriks rotasi untuk notasi (*nutation*).
 P = matriks rotasi untuk presesi (*precession*).

2.4 Sistem Koordinat Orbital

Posisi satelit dalam orbitnya kerap dinyatakan dalam sistem koordinat orbital. Sistem koordinat ini adalah sistem koordinat kartesian dua dimensi, yang titik nolnya berada di pusat bumi, sumbu-sumbu (x dan y) terletak dalam bidang orbit, sumbu x -nya mengarah ke titik *perigee*, dan sumbu y -nya tegak lurus dengan sumbu x (Abidin, 2001), seperti pada gambar 2.3



Gambar 2. 3 Visualisasi Geometrik dari Parameter Orbit GPS (Abidin, 2001)



Gambar 2. 4 Sistem Koordinat Orbital (Abidin, 2001)

Dalam sistem koordinat orbital, dengan mengacu pada gambar 2.4 diatas terlihat bahwa vektor posisi geosentrik satelit $\mathbf{r}(x,y)$ dapat dirumuskan sebagai berikut:

$$x = r \cdot \cos f = a \cdot (\cos E - e) \quad \text{Persamaan 2.3.}$$

$$y = r \cdot \sin f = b \cdot (\sin E - e) \quad \text{Persamaan 2.4.}$$

$$= a \cdot (1 - e^2)^{\frac{1}{2}} \cdot \sin E$$

dimana panjang vector \mathbf{r} dapat diformulasikan sebagai berikut:

$$r = a \cdot (1 - e \cdot \cos E)^{\frac{1}{2}} = \frac{a \cdot (1 - e^2)}{(1 + e \cdot \cos f)} \quad \text{Persamaan 2.5.}$$

Koordinat satelit dalam system orbit ini $\mathbf{r}(x, y, 0)$ selanjutnya dapat ditransformasikan ke sistem koordinat CIS : $\mathbf{X}(X_i, Y_i, Z_i)$ dengan menggunakan hubungan berikut :

$$\mathbf{X} = R_3(-\Omega) \cdot R_1(-i) \cdot R_3(-\omega) \cdot \mathbf{r} \quad \text{Persamaan 2.6.}$$

dimana R_1 dan R_3 adalah matriks rotasi mengelilingi sumbu X dan sumbu Z yang telah diformulasikan seperti:

$$R_1(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix}$$

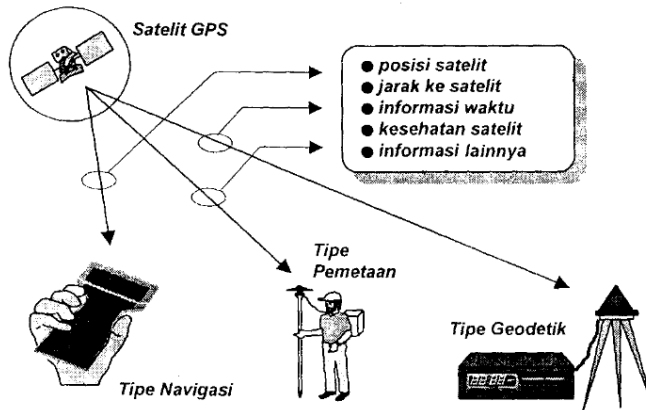
$$R_2(\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix}$$

$$R_3(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Persamaan 2.7.

2.5 Sinyal GPS

Pada prinsipnya satelit GPS memancarkan sinyal – sinyal untuk memberikan informasi posisi satelit tersebut, jarak antara satelit dan penerima beserta informasi waktunya (Abidin, 2007). Sinyal GPS juga menginformasikan kesehatan satelit serta informasi pendukung yang berupa parameter – parameter untuk perhitungan koreksi jam satelit, parameter model ionosfer satu frekuensi, transformasi waktu GPS ke UTC, dan status konstelasi satelit seperti yang ditunjukkan pada gambar 2.5.

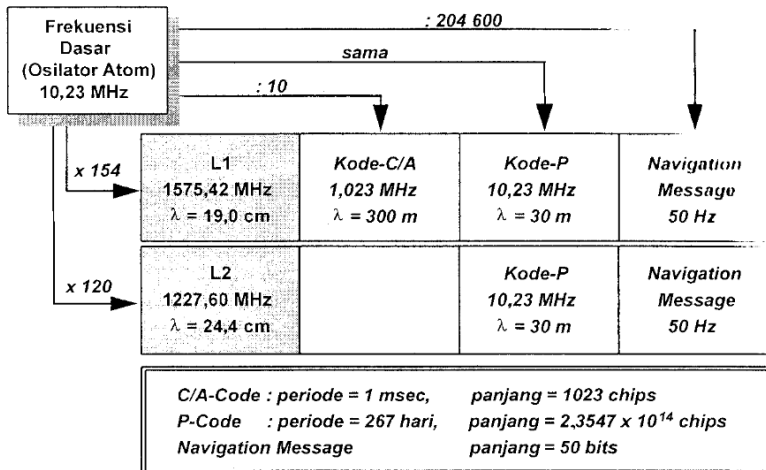


Gambar 2. 5 Informasi pada Sinyal GPS (Abidin, 2007)

Secara umum, sinyal GPS dapat dibagi menjadi 3 komponen, yaitu :
 penginformasian jarak (kode) yang berupa kode-P(Y) dan kode-C/A,

penginformasian posisi satelit (navigation message), dan gelombang pembawa (carrier wave) L1 dan L2.

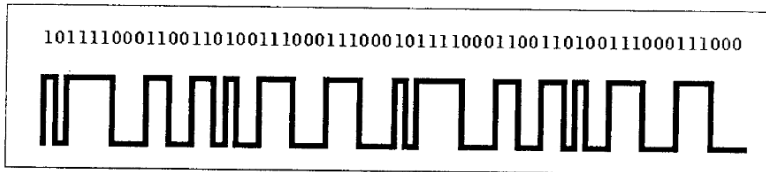
Sinyal GPS menggunakan rentang frekuensi-L dari spektrum gelombang elektromagnetik. Struktur frekuensi dan parameter dasar komponen sinyal GPS ditunjukkan pada gambar 2.6 berikut :



Gambar 2. 6 Struktur dan Parameter Dasar Komponen Sinyal GPS (Abidin, 2007)

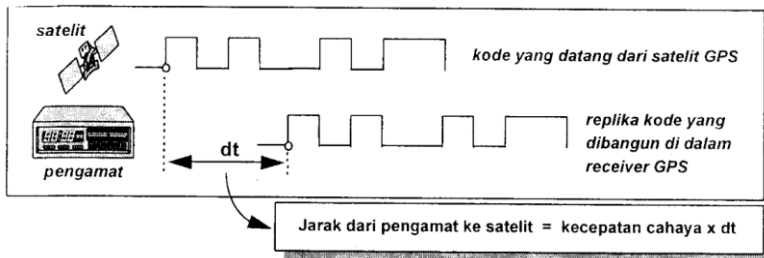
2.5.1 Penginformasian Jarak (Kode)

Ada dua kode pseudo-random noise (PRN) yang dikirimkan oleh satelit GPS dan digunakan sebagai penginformasian jarak, yaitu kode-P (P = Precise atau Private) dan kode-C/A (C/A = Coarse Acquisition atau Clear Access). Kode – kode tersebut merupakan kombinasi bilangan 0 dan 1 (biner) yang ditunjukkan pada gambar 2.7 berikut :



Gambar 2. 7 Contoh Potongan Struktur Kode pada Sinyal GPS (Abidin, 2007)

Kode – kode tersebut sepiintas terlihat seperti rangkaian kombinasi acak angka 0 dan 1, sehingga disebut pseudo-random. Namun kode – kode tersebut punya struktur yang unik dan tertentu yang dibangun dengan suatu algoritma matematis tertentu. Kode-C/A merupakan rangkaian dari 1023 bilangan biner (chips) yang berulang setiap satu milidetik dan hanya dimodulasikan pada gelombang pembawa L1. Sedangkan kode-P merupakan rangkaian bilangan biner yang sangat panjang yaitu $2,3547 \times 10^{14}$ chips dan polanya tidak berulang sampai 267 hari. Kode-P dimodulasikan pada kedua gelombang pembawa L1 dan L2. Dengan mengamati kode-P(Y) ataupun kode-C/A, jarak dari pengamat ke satelit dapat ditentukan. Prinsip pengukurannya adalah dengan membandingkan kode yang diterima dari satelit dengan kode replka yang dimodulasikan di dalam receiver, seperti yang dtunjukkan pada gambar 2.8.



Gambar 2. 8 Prinsip Penentuan Jarak (Abidin, 2007)

Waktu yang diperlukan untuk membandingkan kedua kode (dt) merupakan waktu yang diperlukan oleh kode tersebut untuk menempuh jarak dari satelit ke pengamat. Karena ketelitian jam yang ada pada receiver GPS (jam quartz) berbeda dengan ketelitian jam pada satelit (jam atom), maka jarak yang diukur masih mengandung kesalahan karena ketidaksinkronan waktu antara kedua jam tersebut, sehingga jarak yang diukur disebut jarak semu atau pseudorange.

2.5.2 **Penginformasian Posisi Satelit**

Selain berisi kode – kode, sinyal GPS juga berisi pesan navigasi (navigation message) yang berisi tentang informasi koefisien koreksi jam satelit, parameter orbit, almanak satelit, UTC, parameter koreksi ionosfer, serta informasi spesial lainnya seperti status konstelasi dan kesehatan satelit. Pesan navigasi tersebut ditentukan oleh segmen sistem kontrol dan dikirimkan ke pengguna menggunakan satelit GPS. Salah satu informasi yang terkandung pada pesan navigasi GPS adalah ephemeris (orbit) satelit yang biasa disebut broadcast ephemeris.

Dalam broadcast ephemeris, informasi tentang koordinat satelit tidak diberikan secara langsung dalam koordinat tiga dimensi, tapi dalam bentuk parameter – parameter keplerian orbit GPS yang digunakan untuk menghitung posisi satelit dari waktu ke waktu. Pada umumnya broadcast ephemeris berisi parameter waktu, orbit satelit, dan perturbasi dari orbit satelit. Deskripsi semua parameter tersebut ditunjukkan pada tabel 2.1 berikut.

Tabel 2. 1 Isi dari *Broadcast Ephemeris* GPS (Abidin, 2007)

Parameter Waktu	
t_{oe}	Waktu referensi parameter ephemeris (s)
t_{oc}	Waktu referensi parameter jam (s)
a_o, a_1, a_2	Koefisien polinomial untuk koreksi kesalahan jam satelit, (s, s/s, dan s/s ²)
IOD	<i>Issue of Data</i>
Parameter Orbit Satelit	
\sqrt{a}	Akar dari sumbu panjang ellipsoid (m ^{1/2})
E	Eksentrisitas
i_o	Inklinasi pada waktu t_{oe} (radian)
Ω_o	Bujur dari titik naik (<i>ascending node</i>) pada waktu t_{oe} (radian)
ω	Argumen perigee (radian)
M_o	Anomali menengah pada waktu t_{oe} (radian)
Parameter Perturbasi Orbit	
Δn	Perbedaan nilai <i>mean motion</i> dari nilai hitungannya (radian)
Ω	Kecepatan perubahan dari asensio rekta (radian/s)
\dot{i}	Kecepatan perubahan dari inklinasi (radian/s)
C_{us} dan C_{uc}	Amplitudo dari suku – suku koreksi harmonik sinus dan cosinus untuk argumen lintang (radian)
C_{is} dan C_{ic}	Amplitudo dari suku – suku koreksi harmonik sinus dan cosinus untuk sudut iklinasi (radian)
C_{rs} dan C_{rc}	Amplitudo dari suku – suku koreksi harmonik sinus dan cosinus untuk radius orbit (m)

Dari parameter yang diberikan oleh broadcast ephemeris, dapat ditentukan koordinat satelit pada setiap epoch pengamatan. Algoritma standar yang biasa digunakan untuk penentuan koordinat satelit diberikan pad tabel 2.2 berikut.

Tabel 2. 2 Algoritma Penentuan Koordinat Satelit (Abidin, 2007)

$\mu = 3.986005 \times 10^{14} \text{ m}^3/\text{s}^2$	Nilai konstanta gravitasi bumi (WGS-84)
$\Omega_o = 7.2921151467 \times 10^{-5} \text{ rad/s}$	Kecepatan rotasi bumi (WGS-84)
$\pi = 3.1415926535898$	Nilai π standar untuk GPS
$a = (\sqrt{a})^2$	Nilai sumbu panjang ellipsoid
$n_o = \sqrt{(\mu/a^3)}$	Nilai <i>mean motion</i> nominal
$t_k = t - t_{oe}$	Waktu sejak waktu referensi ephemeris
$n = n_o + \Delta n$	Nilai <i>mean motion</i> yang telah dikoreksi
$M_k = M_o + n.t_k$	Nilai anomali menengah
$\cos \vartheta_k = (\cos E_k - e)/(1 - e.\cos E_k)$	Persamaan untuk menentukan nilai anomali sejati
$\sin \vartheta_k = \sqrt{(1 - e^2)}.\sin E_k/(1 - e.\cos E_k)$	
$\varphi_k = \vartheta_k + \omega$	Nilai argumen lintang
$\delta u_k = C_{uc}.\cos 2\varphi_k + C_{us}.\sin 2\varphi_k$	Nilai koreksi untuk argumen lintang
$\delta r_k = C_{rc}.\cos 2\varphi_k + C_{rs}.\sin 2\varphi_k$	Nilai koreksi untuk radius
$\delta i_k = C_{ic}.\cos 2\varphi_k + C_{is}.\sin 2\varphi_k$	Nilai koreksi untuk inklinasi
$u_k = \varphi_k + \delta u_k$	Nilai argumen lintang yang telah terkoreksi
$r_k = a.(1 - e.\cos E_k) + \delta r_k$	Nilai radius yang telah terkoreksi
$i_k = i_o + \text{idot}.t_k + \delta i_k$	Nilai inklinasi yang telah terkoreksi
$x_k = r_k.\cos u_k$	Koordinat satelit dalam bidang orbit
$y_k = r_k.\sin u_k$	
$\Omega_k = \Omega_o + (\Omega - \Omega_e) t_k - \Omega_e t_{oe}$	Nilai bujur dari titik naik yang telah terkoreksi
$X_k = x_k.\cos \Omega_k - y_k.\sin \Omega_k$	Koordinat geosentrik dari satelit

$Y_k = x_k \cdot \sin \Omega_k + y_k \cdot \cos i_k \cdot \cos \Omega_k$	
$Z_k = y_k \cdot \sin i_k$	

2.6 Data Pengamatan GPS

Data pengamatan dasar GPS adalah waktu tempuh (Δt) dari kode – kode P dan C/A serta fase (*carrier phase*, ϕ) dari gelombang pembawa L1 dan L2. Pada kegiatan pengamatan, dapat diamati sebagian atau seluruh jenis pengamatan tersebut tergantung pada jenis dan tipe alat penerima sinyal GPS yang digunakan (Abidin, 2007). Hasil pengamatan tersebut terkait dengan posisi pengamatan (x,y,z) serta parameter – parameter lainnya melalui hubungan yang diformulasikan sebagai berikut:

$$P_i = \rho + dp + dtrop + dion_i + (dt - dT) + MP_i + \vartheta P_i$$

Persamaan 2.8

$$L_i = \rho + dp + dtrop - dion_i + (dt - dT) + MC_i - \lambda_i \cdot N_i + \vartheta C_i$$

Persamaan 2.9

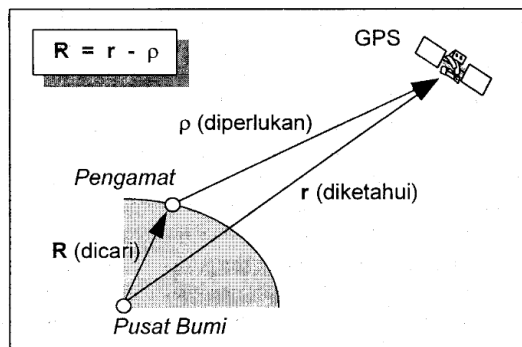
dimana :

- P_i = $c \cdot \Delta t_i$
= pseudorange pada frekuensi f_i (m), ($i = 1,2$)
- L_i = $\lambda_i \cdot \phi_i$
= jarak fase (*carrier range*) pada frekuensi f_i (m), ($i = 1,2$)
- P = jarak geometris antara pengamat (x,y,z) dengan satelit (m)
- c = kecepatan cahaya dalam vakum (m/s)
- λ = panjang gelombang dari sinyal (m)
= c/f (f adalah frekuensi)
- dp = kesalahan jarak yang disebabkan oleh kesalahan ephemeris (orbit)
- $dtrop$ = bias yang disebabkan oleh refraksi troposfer (m)
- $dioni$ = bias yang disebabkan oleh refraksi ionosfer (m) pada frekuensi f_i

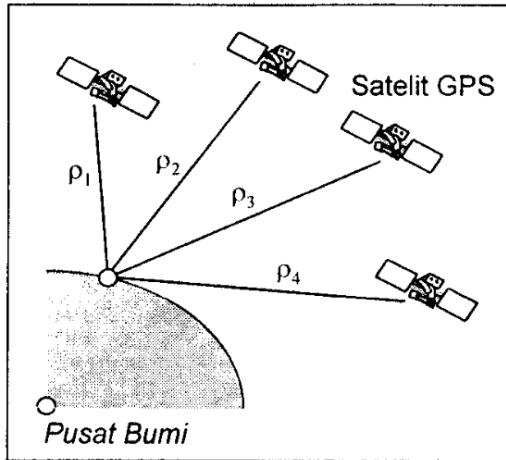
- dt, dT = kesalahan dan offset dari jam receiver dan jam satelit (m)
 MP_i, MC_i = efek *multipath* pada hasil pengamatan P_i dan L_i (m)
 $N1, N2$ = *ambiguitas fase* dari fase sinyal – sinyal $L1$ dan $L2$ (dalam jumlah gelombang)
 $9P_i, 9C_i$ = gangguan (*noise*) pada hasil pengamatan(m)

2.7 Penentuan Posisi Dengan GPS

Pada dasarnya konsep dasar penentuan posisi dengan GPS adalah reseksi (pengikatan ke belakang) dengan jarak, yaitu dengan pengukuran jarak secara simultan ke beberapa satelit GPS (Abidin, et al., 2002). Prinsip dasar penentuan posisi dengan GPS diperlihatkan oleh gambar 2.9. parameter yang akan ditentukan adalah vektor posisi geosentrik pengamat (R). Sedangkan vektor posisi geosentrik satelit GPS (r) diketahui, maka perlu dilakukan pengukuran untuk mendapatkan posisi toposentris satelit terhadap pengamat (ρ). Namun pada pengukuran dengan GPS hanya bisa didapatkan jarak antara satelit dengan pengamat, sehingga rumus pada gambar 2.9 tidak dapat diterapkan. Sehingga perlu dilakukan pengamatan pada beberapa satelit secara simultan, seperti pada gambar 2.10.



Gambar 2. 9 Prinsip Dasar Penentuan Posisi dengan GPS (Abidin, et al., 2002)



Gambar 2. 10 Prinsip Pengamatan Beberapa Satelit (Abidin, et al., 2002)

Pada penentuan koordinat kartesian 3D dari suatu pengamat (X_p, Y_p, Z_p) dengan menggunakan metode penentuan posisi absolut, pengamat pada dasarnya mengukur jarak (ρ) ke beberapa satelit yang sudah diketahui koordinatnya (X_i, Y_i, Z_i) . Seperti pada gambar 2.10, maka ada empat persamaan yang disusun untuk pengamatan jarak ke empat satelit GPS, yaitu :

$$\sqrt{(X_p - X_1)^2 + (Y_p - Y_1)^2 + (Z_p - Z_1)^2} + c \cdot dt = \rho_1$$

$$\sqrt{(X_p - X_2)^2 + (Y_p - Y_2)^2 + (Z_p - Z_2)^2} + c \cdot dt = \rho_2$$

$$\sqrt{(X_p - X_3)^2 + (Y_p - Y_3)^2 + (Z_p - Z_3)^2} + c \cdot dt = \rho_3$$

$$\sqrt{(X_p - X_4)^2 + (Y_p - Y_4)^2 + (Z_p - Z_4)^2} + c \cdot dt = \rho_4$$

Persamaan 2.10.

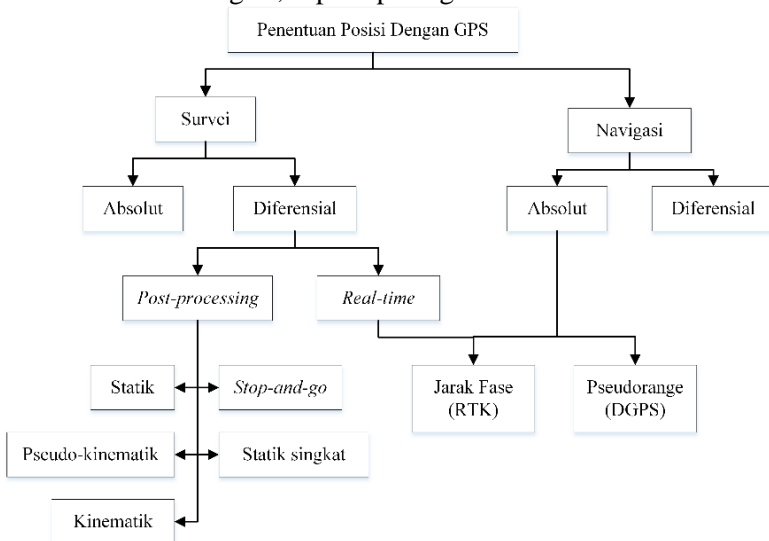
dimana c adalah kecepatan cahaya dan dt adalah kesalahan dan offset dari jam receiver.

Pada persamaan di atas, ada empat parameter yang harus ditentukan/diestimasi yaitu :

1. koordinat pengamat (X_P , Y_P , Z_P), dan
2. kesalahan dan offset dari jam receiver GPS (dt).

Oleh sebab itu, untuk penentuan posisi 3D pada satu epok dengan menggunakan data *pseudorange* diperlukan minimal pengamatan ke empat buah satelit GPS.

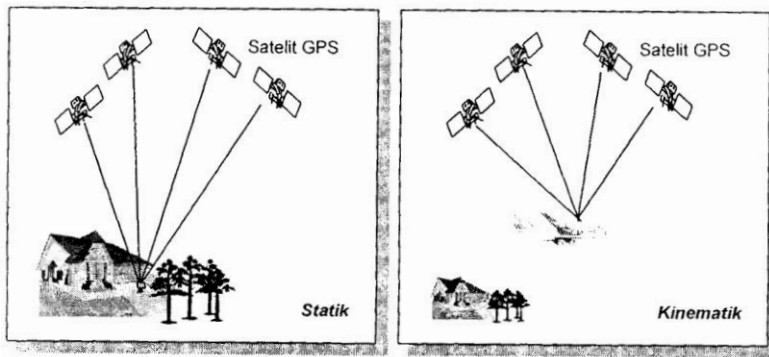
Dalam penentuan posisi dengan GPS, bergantung pada mekanisme pengaplikasiannya, metode penentuan posisi dengan GPS dapat dikelompokkan menjadi beberapa metode, yaitu *absolute*, *differential*, *static*, *rapid static*, *pseudo-kinematic*, dan *stop-and-go*. Sedangkan berdasarkan aplikasinya, juga dapat dibagi menjadi dua kategori, yaitu survei dan navigasi, seperti pada gambar 2.11 berikut:



Gambar 2. 11 Metode Penentuan Posisi dengan GPS (Abidin, et al., 2002)

2.7.1 Penentuan Posisi Absolut

Penentuan posisi secara absolut merupakan metode paling dasar dalam penentuan posisi dengan GPS. Metode ini merupakan metode yang direncanakan pada awalnya oleh pihak militer Amerika untuk memberikan pelayanan navigasi terutama bagi personil dan wahana militer. Metode penentuan posisi ini, dalam moda statik dan kinematik diilustrasikan pada gambar 2.12.



Gambar 2. 12 Metode Penentuan Posisi Absolut (Abidin, 2007)

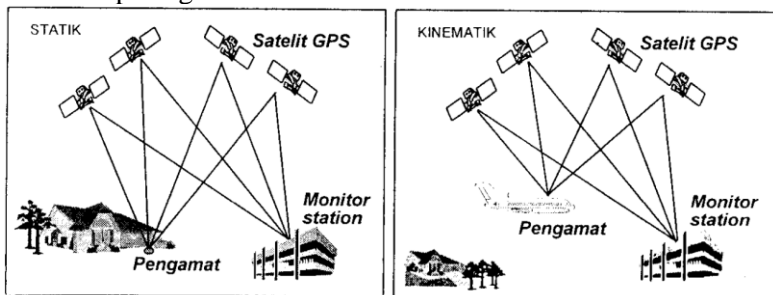
Ada beberapa hal yang terkait dengan penentuan posisi absolut, yaitu :

- a. Metode ini juga disebut *point positioning*, karena penentuan posisi dilakukan per titik tanpa tergantung pada titik lainnya.
- b. Posisi ditentukan dalam sistem WGS-84 terhadap pusat massa Bumi.
- c. Prinsip penentuan posisi adalah reseksi dengan jarak ke beberapa satelit secara simultan.
- d. Untuk penentuan posisi hanya menggunakan satu *receiver* GPS.

- e. Titik yang ditentukan posisinya bisa dalam keadaan diam (statik) maupun dalam keadaan bergerak (kinematik).
- f. Biasanya menggunakan data *pseudorange*. Namun juga terkadang menggunakan data fase dengan mengestimasi ambiguitas fase bersama-sama dengan posisi.
- g. Ketelitian posisi sangat bergantung pada tingkat ketelitian data serta geometri satelit.

2.7.2 Penentuan Posisi Diferensial

Ketelitian pada penentuan posisi secara absolut dapat ditingkatkan dengan menggunakan penentuan posisi secara diferensial. Pada penentuan posisi diferensial, posisi suatu titik ditentukan relatif terhadap titik lainnya yang telah diketahui koordinatnya (*monitor station*). Secara ilustratif metode penentuan posisi ini ditunjukkan pada gambar 2.13.



Gambar 2. 13 Metode Penentuan Posisi Diferensial (Abidin, 2007)

Pada metode diferensial atau yang juga disebut penentuan posisi relatif, dengan mengurangkan data yang diamati oleh dua *receiver* GPS pada yang bersamaan, maka beberapa jenis kesalahan dan bias dari data dapat dieliminasi atau direduksi. Pengeliminasian dan pereduksian ini akan meningkatkan akurasi dan presisi data, dan selanjutnya akan meningkatkan tingkat akurasi

dan presisi dari posisi yang diperoleh. Jenis – jenis kesalahan dan bias yang dapat serta tidak dapat dieliminasi atau direduksi dengan proses pengurangan data ditunjukkan pada tabel 2.3.

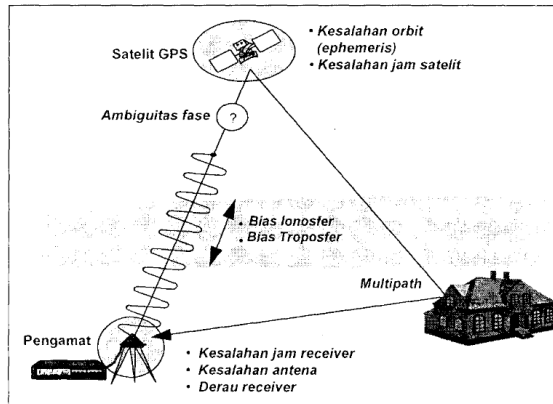
Tabel 2. 3 Efek dari Proses Pengurangan Data (Abidin, 2007)

Kesalahan & Bias	Dampak dari pengurangan data		
	Dapat dieliminasi	Dapat direduksi	Tidak dapat dieliminasi /direduksi
Jam satelit	✓	-	-
Jam receiver	✓	-	-
Orbit (ephemeris)	-	✓	-
Ionosfer	-	✓	-
Troposfer	-	✓	-
Multipath	-	-	✓
Noise	-	-	✓
Selective Availability	✓	✓	-

Efektivitas dari proses pengurangan ini sangat bergantung pada jarak antara *monitor station* dengan titik yang akan ditentukan posisinya. Dalam hal ini, semakin pendek jarak tersebut maka akan semakin efektif dampak dari pengurangan data, dan sebaliknya.

2.8 Kesalahan Dan Bias

Dalam perjalanannya dari satelit hingga mencapai antena di permukaan bumi, sinyal GPS akan dipengaruhi oleh beberapa kesalahan dan bias (Abidin, 2007). Seperti pada gambar 2.14, kesalahan dan bias GPS pada dasarnya dapat dikelompokkan sebagai berikut :

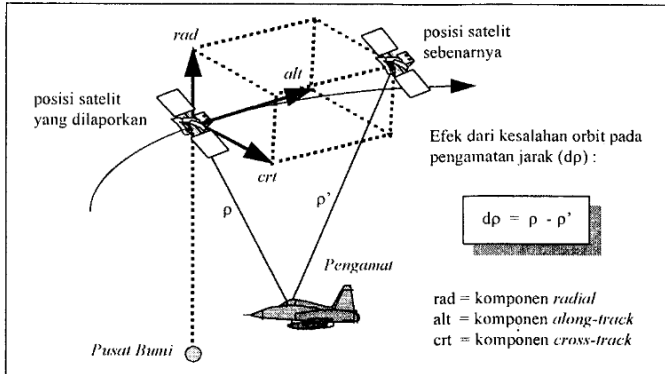


Gambar 2. 14 Kesalahan dan Bias GPS (Abidin, 2007)

1. Satelit, seperti kesalahan ephemeris, jam satelit, dan *selective availability* (SA).
2. Medium propagasi, seperti bias ionosfer dan bias troposfer.
3. Receiver GPS, seperti kesalahan jam receiver, kesalahan yang terkait dengan antenna, dan *noise*.
4. Data pengamatan, seperti ambiguitas fase dan *cycle slips*.
5. Lingkungan sekitar receiver GPS, seperti *multipath* dan *imaging*.

2.8.1 **Kesalahan Ephemeris (Orbit)**

Kesalahan ephemeris (orbit) adalah kesalahan dimana orbit satelit yang dilaporkan oleh ephemeris satelit tidak sama dengan orbit satelit yang sebenarnya. Sehingga posisi satelit yang dilaporkan tidak sama dengan posisi satelit yang sebenarnya seperti pada gambar 2.15 berikut.



Gambar 2. 15 Kesalahan Ephemeris (Orbit) (Abidin, 2007)

Kesalahan orbit satelit GPS pada dasarnya dapat disebabkan oleh tiga faktor berikut secara bersama – sama, yaitu :

- Kekurang telitian pada proses perhitungan orbit satelit oleh satelit – satelit pengontrol satelit.
- Kesalahan dalam prediksi orbit untuk periode waktu setelah *uploading* ke satelit.
- Penerapan kesalahan orbit yang sengaja diterapkan, seperti dalam kasus penerapan *Selective Availability* (SA), yang sejak 2 Mei 2000 telah ditiadakan.

Secara tipikal besar dari setiap komponen kesalahan orbit satelit GPS pada saat ini (tanpa adanya SA) adalah sekitar 2 m (komponen radial), 5 m (komponen *along-track*), dan 3 m (komponen *cross-track*).

Efek kesalahan orbit pada panjang *baseline* dapat ditentukan dengan rumus pendekatan (*rule of thumb*) berikut ini :

$$db = \left(\frac{b}{r} \right) \cdot dr \quad \text{Persamaan 2.11}$$

dimana : dr = besarnya kesalahan orbit
 db = besarnya efek kesalahan orbit

b = panjang *baseline*

r = jarak rata – rata pengamat ke satelit

Pada dasarnya besar kesalahan orbit (dr) tergantung pada jenis ephemeris (orbit) yang digunakan. Ada beberapa jenis informasi orbit satelit GPS dengan nilai kesalahan yang berbeda seperti pada tabel 2.4.

Tabel 2. 4 Nilai Tipikal Kesalahan Orbit GPS (Abidin, 2007)

Ephemeris	Ketelitian	Latency	Update
Almanak	beberapa km	Real time	-
Broadcast (SA Off)	≈ 160 cm	Real time	-
Ultra Rapid (predicted half)	≈ 10 cm	Real time	empat kali sehari
Ultra Rapid (observed half)	< 5 cm	3 jam	empat kali sehari
Rapid	< 5 cm	17 jam	harian
Precise	< 5 cm	≈ 13 hari	mingguan

2.8.2 **Bias Troposfer**

Sinyal dari satelit GPS untuk mencapai antena harus melalui lapisan troposfer, yaitu lapisan atmosfer netral yang berbatasan dengan permukaan bumi. Lapisan troposfer memiliki ketebalan sekitar 9 sampai 16 km, tergantung dengan tempat dan waktu. Efek utama dari troposfer berpengaruh pada kecepatan, atau dengan kata lain terhadap hasil ukuran jarak.

Pada frekuensi sinyal GPS (dibawah 30 GHz), magnitude dari bias troposfer tidak tergantung pada frekuensi, sehingga besarnya tidak dapat diestimasi dengan pengamatan pada dua frekuensi. *Pseudorange* dan fase keduanya diperlambat oleh troposfer dan besar magnitude bias troposfernya sama. Magnitude bias

troposfer berkisar sekitar $\approx 2,3$ m di arah zenith sampai ≈ 20 m pada 10° di atas horizon.

Besar bias troposfer dihitung menggunakan data temperatur, tekanan, dan kelembaban udara yang diukur di permukaan bumi. Terdapat beberapa model standar troposfer, yaitu model Hopfield, Saastamoinen, Black, Marini, dll. Model Saastamoinen dihitung dengan menggunakan persamaan berikut :

$$d_{trop} = \frac{0,002277}{\cos z} \cdot \left\{ p + \left(\frac{1255}{T} + 0,05 \right) \cdot e - \tan^2 z \right\}$$

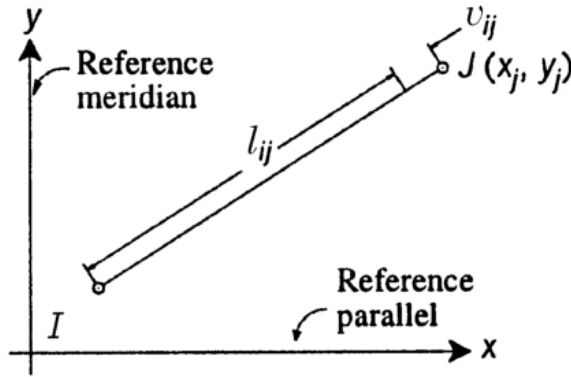
Persamaan 2.12

dimana p adalah tekanan atmosfer (mbar), T adalah temperatur ($^\circ\text{K}$), e adalah tekanan parsial dari uap air (mbar), dan z adalah sudut zenith ke satelit yang diamati.

2.9 Persamaan Pengamatan Jarak

Pada perataan pengukuran trilaterasi menggunakan metode kuadrat terkecil parameter, maka dituliskan persamaan pengamatan yang menunjukkan hubungan antara data pengamatan dan kesalahan acak yang mungkin terjadi untuk mendapatkan nilai yang memungkinkan untuk koordinat x dan y (parameter) dari titik yang berhubungan (Ghilani & Wolf, 2006). Seperti pada gambar 2.16, maka persamaan jaraknya dapat dituliskan seperti berikut

$$l_{ij} + v_{l_{ij}} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \quad \text{Persamaan 2.13.}$$



Gambar 2. 16 Pengamatan Jarak (Ghilani & Wolf, 2006)

Pada persamaan 2.13, l_{ij} merupakan jarak yang diamati antara titik I dan J , v_l adalah residual pengamatan l_{ij} , dan x_i, y_i, x_j, y_j merupakan koordinat yang dianggap benar untuk titik I dan J . Persamaan 2.13 merupakan fungsi nonlinier dengan variabel yang tidak diketahui x_i, y_i, x_j , dan y_j , maka dapat ditulis sebagai berikut :

$$F(x_i, y_i, x_j, y_j) = l_{ij} + v_{l_{ij}} \quad \text{Persamaan 2.14.}$$

dimana

$$F(x_i, y_i, x_j, y_j) = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$$

Persamaan 2.15.

Sistem persamaan nonlinier pada persamaan 2.15 di atas dapat dilinierisasikan dan diselesaikan dengan menggunakan metode pendekatan Taylor. Bentuk linierisasi dari persamaan 2.15 adalah sebagai berikut :

$$F(x_i, y_i, x_j, y_j) = F(x_{io}, y_{io}, x_{jo}, y_{jo}) + \left(\frac{\partial F}{\partial x_i}\right)_o dx_i + \left(\frac{\partial F}{\partial y_i}\right)_o dy_i + \left(\frac{\partial F}{\partial x_j}\right)_o dx_j + \left(\frac{\partial F}{\partial y_j}\right)_o dy_j \quad \text{Persamaan 2.16.}$$

dimana $(\partial F/\partial x_i)_o$, $(\partial F/\partial y_i)_o$, $(\partial F/\partial x_j)_o$, dan $(\partial F/\partial y_j)_o$ merupakan turunan parsial dari F terhadap x_i, y_i, x_j , dan y_j ,

berturut – turut, dievaluasi dengan koordinat pendekatan x_{io} , y_{io} , x_{jo} , dan y_{jo} ; x_i , y_i , x_j , dan y_j adalah parameter yang tidak diketahui; dan dx_i , dy_i , dx_j , dan dy_j adalah koreksi terhadap nilai koordinat pendekatan seperti berikut :

$$\begin{aligned} x_i &= x_{io} + dx_i & y_i &= y_{io} + dy_i & \text{Persamaan 2.17.} \\ x_j &= x_{jo} + dx_j & y_j &= y_{jo} + dy_j \end{aligned}$$

Metode linierisasi dengan pendekatan Taylor ini dapat diterapkan pada persamaan 2.10 dengan parameter yang terdiri dari koordinat pengamat (X_p , Y_p , Z_p) dan kesalahan dan offset dari jam receiver GPS (dt), sedangkan koordinat satelit telah diketahui. Sehingga bentuk linierisasi dari persamaan 2.10 adalah sebagai berikut :

$$\begin{aligned} F(x_p, y_p, z_p) &= F(x_{po}, y_{po}, z_{po}) + \left(\frac{\partial F}{\partial x_p} \right)_o dx_p + \\ &\left(\frac{\partial F}{\partial y_p} \right)_o dy_p + \left(\frac{\partial F}{\partial z_p} \right)_o dz_p \end{aligned} \quad \text{Persamaan 2.18.}$$

dimana

$$F(x_p, y_p, z_p) = \rho_i - c \cdot dt + v_{\rho_i} \quad \text{Persamaan 2.19.}$$

Dengan menggunakan metode turunan parsial, didapatkan persamaan sebagai berikut :

$$\frac{\frac{\partial F}{\partial x_p} = \frac{x_p - x_i}{D_i}}{\frac{z_p - z_i}{D_i}} \quad \frac{\partial F}{\partial y_p} = \frac{y_p - y_i}{D_i} \quad \frac{\partial F}{\partial z_p} = \quad \text{Persamaan 2.20.}$$

dimana

$$D_i = \sqrt{(X_p - X_i)^2 + (Y_p - Y_i)^2 + (Z_p - Z_i)^2}$$

Persamaan 2.21.

Dengan mensubstitusikan persamaan 2.20 pada persamaan 2.18 dan hasilnya disubstitusikan pada persamaan 2.19, maka didapatkan persamaan sebagai berikut :

$$\begin{aligned} &\left(\frac{x_p - x_i}{D_i} \right)_o \partial x_p + \left(\frac{y_p - y_i}{D_i} \right)_o \partial y_p + \left(\frac{z_p - z_i}{D_i} \right)_o \partial z_p + \partial t = \\ &\rho_i - D_{i0} + c \cdot dt + v_{\rho_i} \end{aligned}$$

$$\begin{aligned}
& \left(\frac{X_p - X_2}{D_2} \right)_0 \partial x_p + \left(\frac{Y_p - Y_2}{D_2} \right)_0 \partial y_p + \left(\frac{Z_p - Z_2}{D_2} \right)_0 \partial z_p + \partial t = \\
& \rho_2 - D_{2_0} + c \cdot dt + v_{\rho_2} \\
& \left(\frac{X_p - X_3}{D_3} \right)_0 \partial x_p + \left(\frac{Y_p - Y_3}{D_3} \right)_0 \partial y_p + \left(\frac{Z_p - Z_3}{D_3} \right)_0 \partial z_p + \partial t = \\
& \rho_3 - D_{3_0} + c \cdot dt + v_{\rho_3} \\
& \left(\frac{X_p - X_4}{D_4} \right)_0 \partial x_p + \left(\frac{Y_p - Y_4}{D_4} \right)_0 \partial y_p + \left(\frac{Z_p - Z_4}{D_4} \right)_0 \partial z_p + \partial t = \\
& \rho_4 - D_{4_0} + c \cdot dt + v_{\rho_4}
\end{aligned}$$

Persamaan 2.22.

Persamaan dapat ditunjukkan dalam bentuk matriks sebagai

$$JX = K + V$$

Persamaan 2.23.

dimana J adalah matriks *jacobian* dari turunan parsial, X adalah matriks parameter, K adalah matriks konstanta (jarak *pseudorange* dikurangi jarak yang dihitung dari koordinat pendekatan), dan V adalah matriks residu.

$$\begin{bmatrix}
\left(\frac{X_p - X_1}{D_1} \right)_0 & \left(\frac{Y_p - Y_1}{D_1} \right)_0 & \left(\frac{Z_p - Z_1}{D_1} \right)_0 & 1 \\
\left(\frac{X_p - X_2}{D_2} \right)_0 & \left(\frac{Y_p - Y_2}{D_2} \right)_0 & \left(\frac{Z_p - Z_2}{D_2} \right)_0 & 1 \\
\left(\frac{X_p - X_3}{D_3} \right)_0 & \left(\frac{Y_p - Y_3}{D_3} \right)_0 & \left(\frac{Z_p - Z_3}{D_3} \right)_0 & 1 \\
\left(\frac{X_p - X_4}{D_4} \right)_0 & \left(\frac{Y_p - Y_4}{D_4} \right)_0 & \left(\frac{Z_p - Z_4}{D_4} \right)_0 & 1
\end{bmatrix}
\begin{bmatrix}
\partial x_p \\
\partial y_p \\
\partial z_p \\
\partial t
\end{bmatrix} =
\begin{bmatrix}
\rho_1 - D_{1_0} + c \cdot dt \\
\rho_2 - D_{2_0} + c \cdot dt \\
\rho_3 - D_{3_0} + c \cdot dt \\
\rho_4 - D_{4_0} + c \cdot dt
\end{bmatrix}
+
\begin{bmatrix}
v_{\rho_1} \\
v_{\rho_2} \\
v_{\rho_3} \\
v_{\rho_4}
\end{bmatrix}$$

Persamaan 2.24.

Ukuran baris dari matriks J dan K merupakan jumlah dari pengamatan yang dilakukan pada setiap satelit dan ukuran kolomnya adalah jumlah parameter yang dicari.

2.10 Bahasa Pemrograman C

C merupakan bahasa yang biasa dipakai untuk keperluan pemrograman sistem, antara lain untuk membuat *assembler*, *interpreter*, *editor*, *compiler*, sistem operasi dan program bantu (Kadir, 1997). C digolongkan sebagai bahasa pemrograman kelas menengah. Penggolongannya ke dalam kelas menengah tidak berarti menyatakan bahwa C lebih sulit atau kurang ampuh dibandingkan dengan bahasa berkelas tinggi seperti Pascal dan Basic, namun menunjukkan bahwa C bukanlah bahasa yang berorientasi pada mesin yang merupakan ciri dari bahasa berkelas rendah. C mengkombinasikan elemen dalam bahasa berkelas tinggi dan rendah. Bahasa C dikembangkan di Laboratorium Bell Amerika sekitar tahun 1972 oleh Dennis Ritchie. C merupakan perbaikan dari bahasa B yang disusun oleh Ken Thompson pada tahun 1970. Pada tahun 1978, Brian Kernighan dan Dennis Ritchie menerbitkan buku yang berjudul *The C Programming Language*. Pertama kali C diwujudkan pada komputer DEC PDP-11 yang memakai sistem operasi UNIX.

Kelebihan dari bahasa C adalah sebagai berikut :

1. C memiliki banyak operator untuk manipulasi data.
2. Berbagai struktur data dan pengendalian proses disediakan dalam C, sehingga memungkinkan membuat program yang terstruktur.
3. Dibandingkan dengan bahasa mesin, C jauh lebih mudah dipahami dan pemrogram tidak perlu tahu detail mesin komputer.
4. Kecepatan eksekusi C mendekati kecepatan eksekusi program yang dibuat dengan bahasa berkelas rendah.
5. C memungkinkan memanipulasi data dalam bentuk bit maupun byte secara efisien.

Sedangkan kelemahan dari bahasa C adalah sebagai berikut :

1. Banyaknya operator serta fleksibilitas penulisan program membuat bingung pemakai.
2. Para pemrogram C tingkat pemula umumnya belum pernah mengenal pointer dan tidak biasa menggunakannya.

“Halaman ini sengaja dikosongkan”

BAB III

METODOLOGI

3.1 Peralatan Dan Bahan

3.1.1 Peralatan

Peralatan yang digunakan pada penelitian tugas akhir ini adalah sebagai berikut :

1. Perangkat keras (hardware) :
 - a. Laptop
 - b. GPS Geodetik
2. Perangkat lunak (software) :
 - a. Microsoft Office 2013
 - b. Dev C++

3.1.2 Data

Data yang digunakan pada penelitian ini adalah data rinex navigasi dan rinex observasi hasil pengukuran GPS sebagai berikut :

1. Data pengamatan tanggal 23 November 2015
2. Data pengamatan tanggal 4 Januari 2016
3. Data pengamatan tanggal 5 Januari 2016

3.2 Diagram Alir Metodologi Penelitian

Tahapan pekerjaan yang akan dilaksanakan pada penelitian tugas akhir ini adalah sebagai berikut :



Gambar 3. 1 Diagram Alir Penelitian

Dengan penjelasan dari diagram alir penelitian tersebut adalah sebagai berikut :

1. Identifikasi Masalah

Identifikasi masalah dilakukan untuk menganalisa masalah yang akan diteliti. Permasalahan yang akan diteliti adalah bagaimana membuat suatu program pengolahan data GPS dengan menggunakan bahasa pemrograman C++. Data GPS yang digunakan merupakan data RINEX hasil pengukuran untuk didapatkan koordinat titik dengan menggunakan metode absolut.

2. Studi Literatur

Studi literatur dilakukan untuk mendapatkan berbagai referensi yang mendukung penelitian. Referensi yang dipelajari terkait tentang format data RINEX, perhitungan koordinat satelit, dan perhitungan

koordinat *receiver* berdasarkan data pengamatan beberapa satelit.

3. Desain Program

Desain program dilakukan untuk mendesain menu utama program dan tahapan pemrosesan data oleh program. Desain program juga dilakukan untuk membuat desain awal tampilan program yang akan dibuat.

4. Pembuatan Program

Pembuatan program dilakukan menyusun program pengolahan data GPS berdasarkan desain yang telah dibuat. Pembuatan program dilakukan dengan menggunakan perangkat lunak Dev-C++.

5. Pengujian Program

Pengujian program dilakukan untuk menguji program yang telah dibuat. Program yang telah dibuat akan diuji untuk mengetahui ada atau tidaknya kesalahan (*error*) saat dijalankan dan untuk mengetahui proses pengolahan data akan memberikan hasil yang baik atau tidak.

6. Perbaikan Program

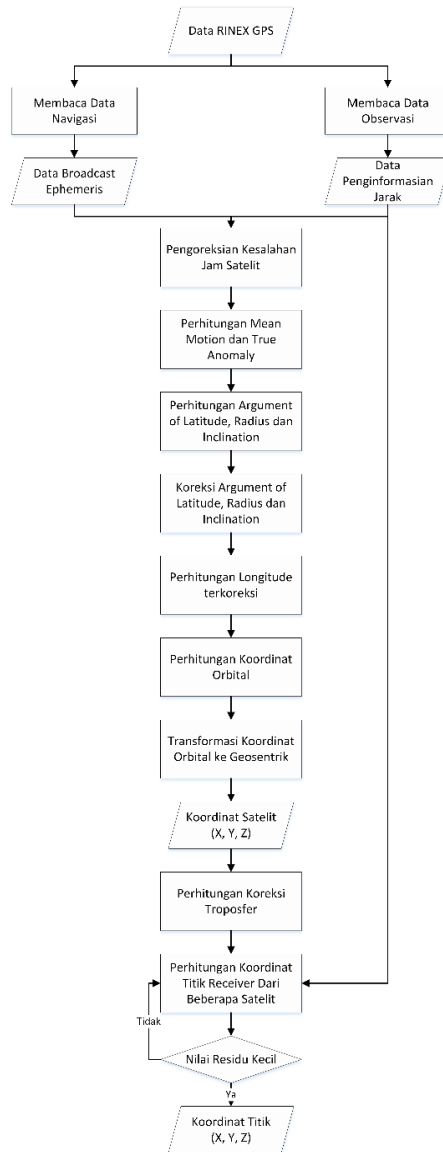
Pembuatan program dilakukan untuk memperbaiki *error* yang terjadi saat program dijalankan atau *error* pada algoritma dan fungsi yang digunakan yang mengakibatkan hasil pengolahan data tidak sesuai.

7. Penyusunan Laporan

Penyusunan laporan merupakan tahap akhir dari penelitian tugas akhir ini. Penyusunan laporan dilakukan untuk menunjukkan hasil penelitian agar dapat memberikan manfaat dan diketahui oleh pihak lain.

3.3 Diagram Alir Pengolahan Data

Tahapan pengolahan data yang akan dijalankan pada program ini adalah sebagai berikut :



Gambar 3. 2 Diagram Alir Pengolahan Data

Dengan penjelasan dari diagram alir pengolahan data tersebut adalah sebagai berikut :

1. Membaca data RINEX

Program akan membuka data pesan navigasi satelit yang menyimpan informasi *broadcast ephemeris* yang terdiri dari parameter waktu, parameter orbit satelit dan parameter perturbasi orbit seperti pada tabel 1, yang akan digunakan untuk mengetahui koordinat satelit – satelit yang terhubung dengan *receiver*. Program juga akan membaca data observasi yang menyimpan informasi hasil pengamatan dari satelit ke pengamat. Informasi tersebut terdiri dari informasi waktu pengamatan, *pseudorange*, fase, dan frekuensi doppler.

2. Perhitungan koordinat satelit

Selanjutnya program akan melakukan perhitungan koordinat satelit berdasarkan data *broadcast ephemeris* yang telah didapatkan. Adapun tahap – tahap perhitungan koordinat satelit adalah sebagai berikut :

a. Pengoreksian kesalahan jam satelit

- Menghitung julian date dari epok yang akan ditentukan nilai koordinatnya

$$JD = \text{floor}(365.25 \times \text{Year}) \\ + (\text{floor}(30.6001 \times (\text{Month} + 1)) + \text{Day} + (UT/24) \\ + 1720981.5$$

- Menghitung *Week* dan *Second of Week* dari data epok

$$e \\ = \text{floor} \left(\frac{\text{floor}(JD + 0.5) + 1537 - 122.1}{365.25} \right) \\ f = \text{floor} \left(\frac{\text{floor}(JD + 0.5) + 1537 - e}{30.6001} \right) \\ d = \left(\frac{\text{floor}(JD + 0.5) + 1537 - e}{365.25} \right)$$

$$\text{Day of Week} = \text{Rem}(\text{floor}(\text{JD} + 0.5), 7)$$

$$\text{Week} = \text{floor}\left(\frac{\text{JD} - 2444244.5}{7}\right)$$

$$T_{rx} = \text{Rem}((d + \text{Day of Week} + 1), 86400)$$

- Menghitung koreksi waktu dari epok terhadap kesalahan jam satelit. Koreksi waktu ini diperlukan karena meskipun semua satelit memiliki sistem jam atom, akan tetapi memiliki kekurangan dalam hal sinkronisasi antara *broadcast* sinyal satelit dan sistem waktu GPS

$$T_s = T_{rx} - (P/c)$$

$$dt = T_s - T_{oc}$$

$$\varepsilon^{sc} = a_0 + a_1 x dt + a_2 x dt^2$$

$$T_{GPS} = T_s - \varepsilon^{sc}$$

$$T_K = T_{GPS} - T_{oe}$$

- b. Perhitungan mean motion dan true anomaly
 - Menghitung nilai *Mean Motion* yang telah terkoreksi

$$n = n_0 + \Delta n$$
 - Menghitung nilai *Mean Anomaly*

$$M_k = M_0 + n \cdot t_k$$
 - Menghitung nilai *True Anomaly*

$$\cos \vartheta k = \frac{\cos Ek - e}{1 - e \cdot \cos Ek}$$

$$\sin \vartheta k = \sqrt{(1 - e^2)} \cdot \sin Ek / (1 - e \cdot \cos Ek)$$
- c. Perhitungan argument of latitude, radius, dan inclination
 - Menghitung nilai *Argument of Latitude*

$$\varphi k = \vartheta k + \omega$$
 - Menghitung koreksi *Argument of Latitude*

$$\delta uk = C_{uc} \cdot \cos 2\varphi k + C_{us} \cdot \sin 2\varphi k$$
 - Menghitung koreksi *Radius*

$$\delta rk = C_{rc} \cdot \cos 2\varphi k + C_{rs} \cdot \sin 2\varphi k$$
 - Menghitung koreksi *Inclination*

- $$\delta ik = Cic.\cos 2\phi k + Cis.\sin 2\phi k$$
- Menghitung *Argument of Latitude* terkoreksi

$$uk = \phi k + \delta uk$$
 - Menghitung *Radius* terkoreksi

$$rk = a.(1 - e.\cos Ek) + \delta rk$$
 - Menghitung *Inclination* terkoreksi

$$ik = io + idot.tk + \delta ik$$
- d. Perhitungan longitude terkoreksi

$$\Omega k = \Omega o + (\Omega - \Omega e) tk - \Omega e toe$$
- e. Perhitungan koordinat orbital

$$xk = rk.\cos uk$$

$$yk = rk.\sin uk$$
- f. Transformasi koordinat orbital ke geosentrik

$$Xk = xk.\cos \Omega k - yk.\sin ik.\sin \Omega k$$

$$Yk = xk.\sin \Omega k + yk.\cos ik.\cos \Omega k$$

$$Zk = yk.\sin ik$$
3. Perhitungan koreksi troposfer
 Bias troposfer dihitung menggunakan model Saastomeinen, dimana p adalah tekanan atmosfer, T adalah temperatur, e adalah tekanan parsial dari uap air, dan z adalah sudut zenith ke satelit yang diamati.
- $$d_{trop} = \frac{0.002277}{\cos z} \left\{ p + \left(\frac{1255}{T} + 0.05 \right) e - \tan^2 z \right\}$$
4. Perhitungan koordinat receiver
 Perhitungan koordinat receiver dilakukan dengan menggunakan data pengamatan jarak atau *pseudorange* yang diamati dari beberapa satelit. Persamaan dasar penentuan posisinya adalah sebagai berikut :

$$\sqrt{(X_p - X_1)^2 + (Y_p - Y_1)^2 + (Z_p - Z_1)^2} + c.dt = \rho_1$$

$$\sqrt{(X_p - X_2)^2 + (Y_p - Y_2)^2 + (Z_p - Z_2)^2} + c.dt = \rho_2$$

$$\sqrt{(X_p - X_3)^2 + (Y_p - Y_3)^2 + (Z_p - Z_3)^2} + c.dt = \rho_3$$

$$\sqrt{(X_p - X_4)^2 + (Y_p - Y_4)^2 + (Z_p - Z_4)^2} + c \cdot dt = \rho_4$$

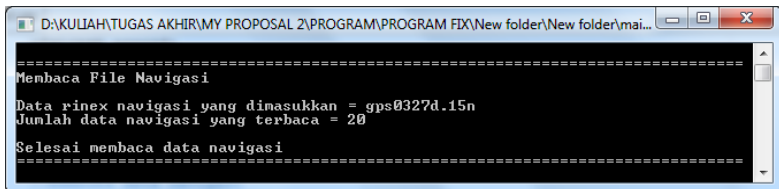
Terdapat 4 parameter yang akan dicari, yaitu dX, dY, dZ, dan dt, sehingga dibutuhkan minimal pengamatan pada 4 satelit.

BAB IV

HASIL DAN PEMBAHASAN

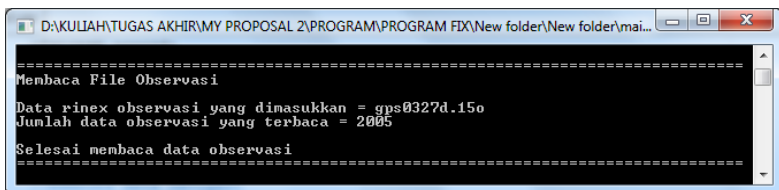
4.1 Tampilan Program Pengolahan Data GPS

Program pengolahan data pengukuran GPS pada penelitian ini dibuat dengan menggunakan bahasa pemrograman C++. Perangkat lunak yang digunakan untuk proses pemrograman adalah Dev C++ versi 4.9.9.2 milik Bloodshed yang merupakan perangkat lunak *Open Source*. Program ini melakukan pengolahan data GPS dengan empat tahap utama, yaitu tahap pembacaan data RINEX navigasi dan observasi, perhitungan koordinat satelit, perhitungan koreksi troposfer, dan perhitungan koordinat titik pengamatan. Program ini digunakan untuk menentukan koordinat titik pengamatan dengan menggunakan metode pengukuran absolut. Berikut adalah tampilan dari program yang telah dibuat.



Gambar 4. 1 Proses Pembacaan Data Navigasi

Gambar 4.1 menunjukkan tahap pembacaan data RINEX navigasi. Program akan menampilkan nama dari data navigasi yang dibaca dan jumlah data ephemeris satelit yang terekam.



Gambar 4. 2 Proses Pembacaan Data Observasi

Gambar 4.2 menunjukkan tahap pembacaan data RINEX observasi. Program akan menampilkan nama dari data observasi yang dibaca dan jumlah epok pengamatan yang terdapat pada data observasi tersebut .

```

D:\KULIAH\TUGAS AKHIR\MY PROPOSAL 2\PROGRAM\PROGRAM FIX\New folder\New folder\mai...
Hasil perhitungan koordinat geosentrik titik receiver :
X = -2451544.527 m
Y = 5832786.186 m
Z = -803235.598 m

Hasil perhitungan koordinat geografis titik receiver :
Lintang = -7.283393406 derajat
Bujur = 112.797219937 derajat
Tinggi = 28.950 m

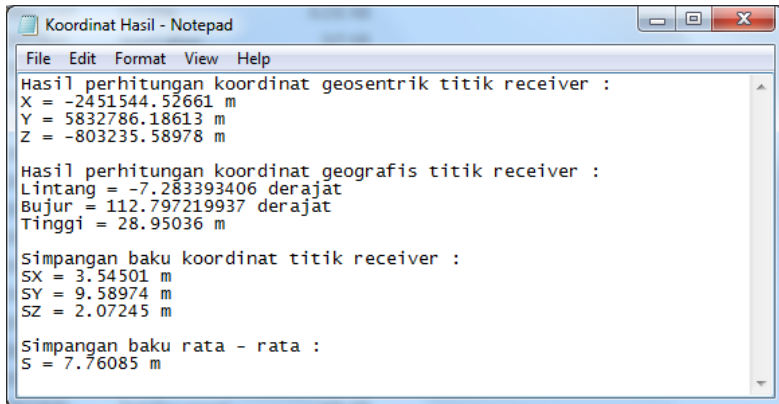
Simpangan baku rata - rata :
S = 7.76885 m
  
```

Gambar 4. 3 Penampilan Hasil Pengolahan Data

Gambar 4.3 akan menunjukkan hasil dari pengolahan data navigasi dan observasi. Program akan menampilkan nilai koordinat titik pengamatan yang ditunjukkan dalam sistem koordinat kartesian geosentrik dan sistem koordinat geografis serta ketinggiannya. Juga akan ditampilkan nilai dari simpangan baku rata – rata rdari hasil perhitungan koordinat titik pengamatan. Program juga akan menghasilkan 2 data yang disimpan dalam format .txt sebagai berikut.

File	Edit	Format	View	Help			
-2451539.21124	5832765.21912	-803232.91231	4.82537	7.39484	-7.283393910	112.797249133	1
-2451540.05902	5832767.01608	-803232.75893	5.11545	9.34445	-7.283390259	112.797249905	2
-2451539.96672	5832767.25841	-803233.17343	5.28756	9.58313	-7.283393762	112.797248284	3
-2451540.15790	5832767.06466	-803233.10264	5.30982	9.47046	-7.283393246	112.797250560	4
-2451540.07506	5832767.76264	-803232.88272	5.02178	10.04900	-7.283390573	112.797247419	5
-2451540.35302	5832767.93372	-803232.82041	5.17502	10.30437	-7.283389710	112.797249139	6
-2451540.57711	5832767.44181	-803232.44948	5.20701	9.89365	-7.283386803	112.797252736	7
-2451540.52519	5832767.75437	-803232.53603	5.43416	10.17049	-7.283387272	112.797251206	8
-2451540.62152	5832767.39482	-803232.33741	5.17358	9.85355	-7.283385828	112.797253272	9
-2451540.68291	5832766.59133	-803232.14989	4.97786	9.11862	-7.283384968	112.797256604	10
-2451540.95931	5832766.99371	-803231.61101	4.82067	9.52449	-7.283379587	112.797257499	11
-2451540.83687	5832766.88945	-803231.84610	4.81281	9.41189	-7.283384860	112.797256843	12
-2451540.88047	5832767.12009	-803231.93940	4.51908	9.65139	-7.283382433	112.797256398	13
-2451540.60901	5832766.49783	-803231.96677	4.52484	8.98150	-7.283383457	112.797256315	14
-2451540.41198	5832765.99323	-803231.61471	4.68015	8.39971	-7.283380920	112.797256440	15
-2451540.16223	5832765.17141	-803231.33722	4.87976	7.51704	-7.283379411	112.797257239	16
-2451540.31026	5832766.79534	-803231.86143	4.63225	9.12538	-7.283382331	112.797252777	17
-2451540.63344	5832767.28947	-803232.17592	4.72141	9.72303	-7.283384507	112.797253811	18
-2451540.34765	5832766.56822	-803232.00880	4.65054	8.95075	-7.283383876	112.797253886	19
-2451540.44147	5832766.41873	-803232.12167	4.73336	8.86441	-7.283385005	112.797255194	20
-2451540.32355	5832766.58254	-803232.17848	4.69654	8.97609	-7.283385393	112.797253634	21
-2451540.33620	5832767.04496	-803231.92776	4.78531	9.37202	-7.283382650	112.797252117	22

Gambar 4. 4 Data Koordinat Tiap Epok



Gambar 4. 5 Hasil Perhitungan Koordinat Titik Pengamatan

Data yang ditunjukkan pada Gambar 4.4 merupakan data hasil perhitungan koordinat titik pengamatan dari setiap epok pengamatan. Data tersebut terdiri dari data koordinat geosentrik (X, Y, Z), simpangan baku (S), tinggi, koordinat geografi (lintang dan bujur), serta nomor epok. Sedangkan data pada Gambar 4.5 merupakan data hasil akhir koordinat titik pengamatan yang dihitung dengan merata – rata nilai koordinat per epok. Data yang disimpan adalah koordinat geosentrik (X, Y, Z), simpangan baku dari koordinat geosentrik (SX, SY, SZ), simpangan baku rata – rata, dan koordinat geografis (lintang, bujur, dan tinggi).

4.2 Hasil Pengolahan Data Pengamatan Tanggal 23 November 2015

Pengolahan data pengamatan pada tanggal 23 November 2015 dilakukan dengan menggunakan program yang telah dibuat. Dengan mempergunakan data RINEX yang didapatkan dari pengukuran, program akan melakukan perhitungan koordinat titik yang diamati. Pada perhitungan koordinat ini dilakukan koreksi troposfer untuk mengurangi efek bias troposfer. Pada Tabel 4.1 ditunjukkan hasil perhitungan

koordinat tanpa menggunakan koreksi troposfer dan Tabel 4.2 menunjukkan hasil perhitungan dengan menggunakan koreksi troposfer. Pada hasil perhitungan dengan menggunakan koreksi troposfer memberikan nilai simpangan baku (S) lebih kecil dari perhitungan tanpa koreksi pada setiap sinyal yang digunakan. Nilai simpangan baku tersebut menunjukkan besarnya nilai presisi dari data koordinat titik pengamatan yang didapatkan dari perhitungan koordinat pada tiap epok. Pada Tabel 4.1 diperoleh bahwa sinyal C1 memberikan nilai simpangan baku paling kecil dan P2 memberikan nilai simpangan baku paling besar. Sama seperti Tabel 4.1, pada Tabel 4.2 simpangan baku terkecil pada sinyal C1 dan terbesar pada sinyal P2.

Tabel 4. 1 Hasil Perhitungan Tanpa Koreksi Troposfer

Sinyal	X (m)	Y (m)	Z (m)	S (m)
C1	-2451553.250	5832811.347	-803240.663	6.244
P1	-2451552.831	5832811.292	-803240.537	6.732
P2	-2451555.624	5832819.817	-803241.757	8.232

Tabel 4. 2 Hasil Perhitungan Dengan Koreksi Troposfer

Sinyal	X (m)	Y (m)	Z (m)	S (m)
C1	-2451541.254	5832776.738	-803234.556	6.085
P1	-2451541.733	5832777.660	-803234.370	6.414
P2	-2451544.527	5832786.186	-803235.590	7.761

Pada Tabel 4.3 dan 4.4 ditunjukan selisih antara nilai koordinat titik pengamatan hasil perhitungan tanpa dan menggunakan koreksi troposfer terhadap nilai koordinat titik pengamatan yang dianggap benar dari hasil pengolahan data menggunakan perangkat lunak lain. Nilai koordinat tersebut adalah sebagai berikut :

X = -2451544.694 m

Y = 5832792.414 m

Z = -803237.449 m

Berdasarkan selisih nilai koordinat yang dianggap benar dengan koordinat hasil pengolahan data pada setiap sinyal yang digunakan, didapatkan nilai RMSE (*Root Mean Square Error*) yang menunjukkan keakurasian hasil perhitungan koordinat. Didapatkan bahwa koordinat hasil perhitungan dengan menggunakan koreksi troposfer memiliki nilai RMSE yang lebih kecil daripada koordinat hasil perhitungan tanpa menggunakan koreksi troposfer. Pada Tabel 4.3 diperoleh nilai RMSE paling kecil pada sinyal P1 dan paling besar pada sinyal P2. Sedangkan pada Tabel 4.4, nilai RMSE paling kecil didapatkan pada sinyal P2 dan paling besar pada sinyal C1.

Tabel 4. 3 Keakuratan Hasil Koordinat Tanpa Koreksi Troposfer

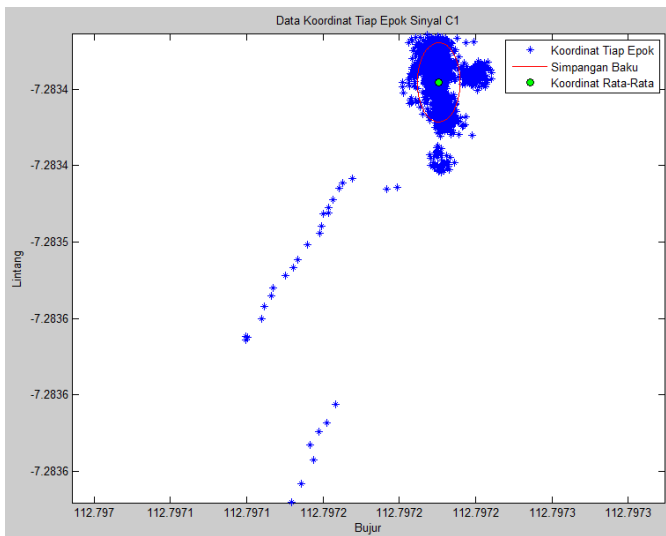
Sinyal	dX (m)	dY (m)	dZ (m)	RMSE (m)
C1	8.556	-18.933	3.214	22.345
P1	8.137	-18.878	3.088	21.446
P2	10.930	-27.403	4.308	30.314

Tabel 4. 4 Keakuratan Hasil Koordinat Dengan Koreksi Troposfer

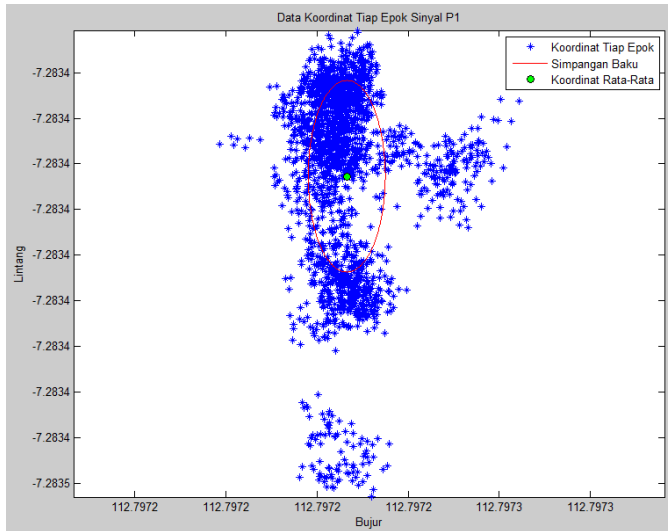
Sinyal	dX (m)	dY (m)	dZ (m)	RMSE (m)
C1	-3.440	15.676	-2.893	20.019
P1	-2.961	14.754	-3.079	18.663
P2	-0.167	6.228	-1.859	12.292

Nilai koordinat titik yang didapatkan pada tabel 4.1 dan 4.2 tersebut merupakan nilai rata – rata dari nilai koordinat titik pada tiap epok pengamatan. Data – data koordinat pada tiap epok tersebut selanjutnya dijumlahkan kemudian dirata – rata sehingga mendapatkan nilai koordinat titik pengamatan. Sebaran nilai koordinat titik pada tiap epok dengan menggunakan koreksi troposfer ditunjukkan pada gambar berikut. Gambar 4.6 menunjukkan sebaran titik koordinat yang dihitung dengan menggunakan data *pseudorange* pada sinyal C1. Gambar 4.7 menunjukkan sebaran titik koordinat yang dihitung dengan menggunakan data sinyal P1. Gambar 4.8

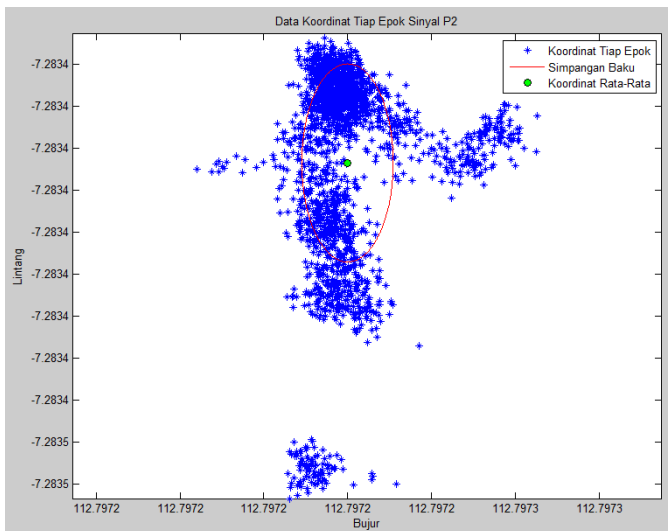
menunjukkan sebaran titik koordinat yang dihitung dengan menggunakan data sinyal P2. Titik berwarna biru menunjukkan sebaran dari koordinat hasil perhitungan pada setiap epok. Titik berwarna hijau menunjukkan koordinat titik pengamatan hasil rata – rata dari koordinat tiap epok. Dan garis berwarna merah menunjukkan simpangan baku dari hasil perhitungan, dimana titik yang berada di dalam garis merah menunjukkan bahwa titik tersebut dapat diterima.



Gambar 4. 6 Sebaran Koordinat Tiap Epok pada Sinyal C1



Gambar 4. 7 Sebaran Koordinat Tiap Epok pada Sinyal P1



Gambar 4. 8 Sebaran Koordinat Tiap Epok pada Sinyal P2

4.3 Hasil Pengolahan Data Pengamatan Tanggal 4 Januari 2016

Pengolahan data pengamatan pada tanggal 4 Januari 2015 dilakukan dengan menggunakan program yang telah dibuat. Tabel 4.5 ditunjukkan hasil perhitungan koordinat tanpa menggunakan koreksi troposfer dan Tabel 4.6 menunjukkan hasil perhitungan dengan menggunakan koreksi troposfer. Pada hasil perhitungan dengan menggunakan koreksi troposfer memberikan nilai simpangan baku (S) lebih kecil dari perhitungan tanpa koreksi pada setiap sinyal yang digunakan. Nilai simpangan baku tersebut menunjukkan besarnya nilai presisi dari data koordinat titik pengamatan yang didapatkan dari perhitungan koordinat pada tiap epok. Pada Tabel 4.5 diperoleh bahwa sinyal P1 memberikan nilai simpangan baku paling kecil dan P2 memberikan nilai simpangan baku paling besar. Sama seperti Tabel 4.5, pada Tabel 4.6 simpangan baku terkecil pada sinyal P1 dan terbesar pada sinyal P2.

Tabel 4. 5 Hasil Perhitungan Tanpa Koreksi Troposfer

Sinyal	X (m)	Y (m)	Z (m)	S (m)
C1	-2426332.474	5838240.471	-846365.43	6.030
P1	-2426332.356	5838239.945	-846365.384	5.982
P2	-2426333.905	5838244.023	-846365.857	7.385

Tabel 4. 6 Hasil Perhitungan Dengan Koreksi Troposfer

Sinyal	X (m)	Y (m)	Z (m)	S (m)
C1	-2426316.953	5838204.65	-846359.674	5.105
P1	-2426317.043	5838205.171	-846360.015	4.993
P2	-2426318.591	5838209.249	-846360.488	6.216

Pada Tabel 4.7 dan 4.8 ditunjukan selisih antara nilai koordinat titik pengamatan hasil perhitungan tanpa dan menggunakan koreksi troposfer terhadap nilai koordinat titik pengamatan yang dianggap benar dari hasil pengolahan data

menggunakan perangkat lunak lain. Nilai koordinat tersebut adalah sebagai berikut :

$$X = -2426324.189 \text{ m}$$

$$Y = 5838222.058 \text{ m}$$

$$Z = -846362.296 \text{ m}$$

Berdasarkan selisih nilai koordinat yang dianggap benar dengan koordinat hasil pengolahan data pada setiap sinyal yang digunakan, didapatkan nilai RMSE yang menunjukkan keakurasian hasil perhitungan koordinat. Didapatkan bahwa koordinat hasil perhitungan dengan menggunakan koreksi troposfer memiliki nilai RMSE yang lebih kecil daripada koordinat hasil perhitungan tanpa menggunakan koreksi troposfer. Pada Tabel 4.7 diperoleh nilai RMSE paling kecil pada sinyal P1 dan paling besar pada sinyal P2. Sedangkan pada Tabel 4.8, nilai RMSE paling kecil didapatkan pada sinyal P2 dan paling besar pada sinyal C1.

Tabel 4. 7 Keakuratan Hasil Koordinat Tanpa Koreksi Troposfer

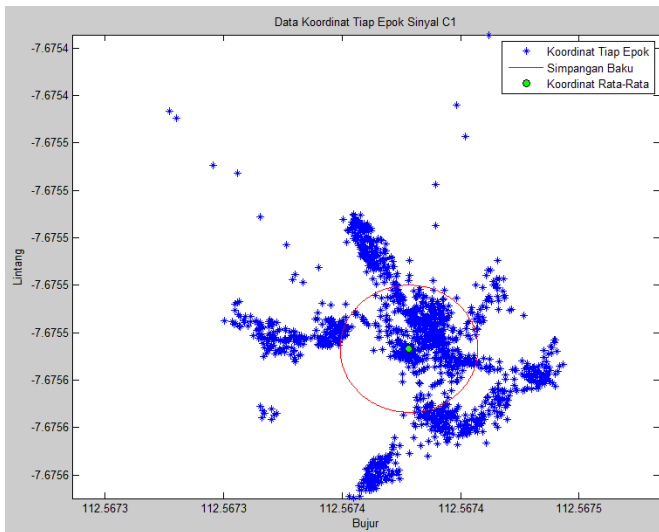
Sinyal	dX (m)	dY (m)	dZ (m)	RMSE (m)
C1	8.285	-18.413	3.142	23.703
P1	8.167	-17.887	3.088	23.678
P2	9.716	-21.965	3.561	28.137

Tabel 4. 8 Keakuratan Hasil Koordinat Dengan Koreksi Troposfer

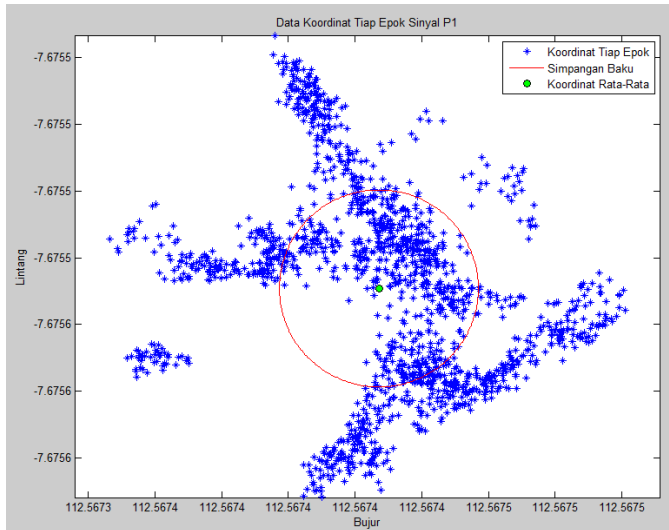
Sinyal	dX (m)	dY (m)	dZ (m)	RMSE (m)
C1	-7.236	17.405	-2.622	21.542
P1	-7.146	16.887	-2.281	20.992
P2	-5.598	12.810	-1.808	17.828

Nilai koordinat titik yang didapatkan pada tabel 4.5 dan 4.6 tersebut merupakan nilai rata – rata dari nilai koordinat titik pada tiap epok pengamatan. Data – data koordinat pada tiap epok tersebut selanjutnya dijumlahkan kemudian dirata – rata sehingga mendapatkan nilai koordinat titik pengamatan. Sebaran nilai koordinat titik pada tiap epok dengan

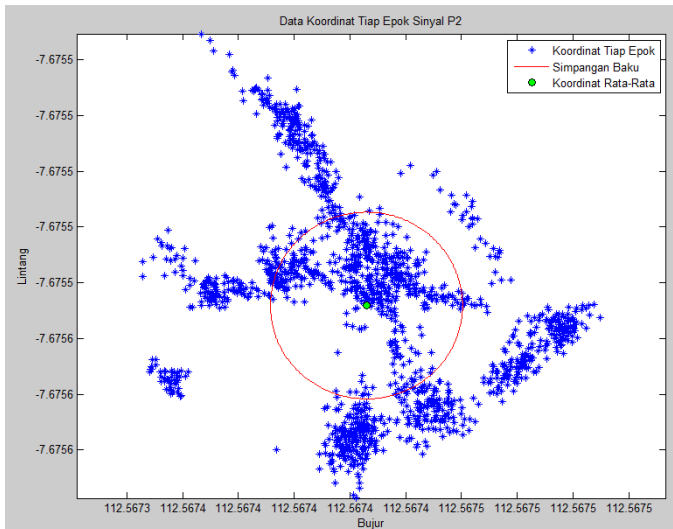
menggunakan koreksi troposfer ditunjukkan pada gambar berikut. Gambar 4.9 menunjukkan sebaran titik koordinat yang dihitung dengan menggunakan data pseudorange pada sinyal C1. Gambar 4.10 menunjukkan sebaran titik koordinat yang dihitung dengan menggunakan data sinyal P1. Gambar 4.11 menunjukkan sebaran titik koordinat yang dihitung dengan menggunakan data sinyal P2. Titik berwarna biru menunjukkan sebaran dari koordinat hasil perhitungan pada setiap epok. Titik berwarna hijau menunjukkan koordinat titik pengamatan hasil rata – rata dari koordinat tiap epok. Dan garis berwarna merah menunjukkan simpangan baku dari hasil perhitungan, dimana titik yang berada di dalam garis merah menunjukkan bahwa titik tersebut dapat diterima.



Gambar 4. 9 Sebaran Koordinat Tiap Epok pada Sinyal C1



Gambar 4. 10 Sebaran Koordinat Tiap Epok pada Sinyal P1



Gambar 4. 11 Sebaran Koordinat Tiap Epok pada Sinyal P2

4.4 Hasil Pengolahan Data Pengamatan Tanggal 5 Januari 2016

Pengolahan data pengamatan pada tanggal 5 Januari 2015 dilakukan dengan menggunakan program yang telah dibuat. Tabel 4.9 ditunjukkan hasil perhitungan koordinat tanpa menggunakan koreksi troposfer dan Tabel 4.10 menunjukkan hasil perhitungan dengan menggunakan koreksi troposfer. Pada hasil perhitungan dengan menggunakan koreksi troposfer memberikan nilai simpangan baku (S) lebih kecil dari perhitungan tanpa koreksi pada setiap sinyal yang digunakan. Nilai simpangan baku tersebut menunjukkan besarnya nilai presisi dari data koordinat titik pengamatan yang didapatkan dari perhitungan koordinat pada tiap epok. Pada Tabel 4.9 diperoleh bahwa sinyal P1 memberikan nilai simpangan baku paling kecil dan P2 memberikan nilai simpangan baku paling besar. Sama seperti Tabel 4.9, pada Tabel 4.10 simpangan baku terkecil pada sinyal P1 dan terbesar pada sinyal P2.

Tabel 4. 9 Hasil Perhitungan Tanpa Koreksi Troposfer

Sinyal	X (m)	Y (m)	Z (m)	S (m)
C1	-2426332.794	5838239.686	-846365.662	6.564
P1	-2426332.857	5838239.075	-846365.403	6.409
P2	-2426334.973	5838243.584	-846366.007	7.884

Tabel 4. 10 Hasil Perhitungan Dengan Koreksi Troposfer

Sinyal	X (m)	Y (m)	Z (m)	S (m)
C1	-2426317.395	5838204.066	-846359.782	5.475
P1	-2426317.654	5838204.627	-846359.986	5.299
P2	-2426319.771	5838209.136	-846360.590	6.601

Pada Tabel 4.11 dan 4.12 ditunjukkan selisih antara nilai koordinat titik pengamatan hasil perhitungan tanpa dan menggunakan koreksi troposfer terhadap nilai koordinat titik pengamatan yang dianggap benar dari hasil pengolahan data

menggunakan perangkat lunak lain. Nilai koordinat tersebut adalah sebagai berikut :

$$X = -2426323.116 \text{ m}$$

$$Y = 5838220.327 \text{ m}$$

$$Z = -846362.068 \text{ m}$$

Berdasarkan selisih nilai koordinat yang dianggap benar dengan koordinat hasil pengolahan data pada setiap sinyal yang digunakan, didapatkan nilai RMSE yang menunjukkan keakurasian hasil perhitungan koordinat. Didapatkan bahwa koordinat hasil perhitungan dengan menggunakan koreksi troposfer memiliki nilai RMSE yang lebih kecil daripada koordinat hasil perhitungan tanpa menggunakan koreksi troposfer. Pada Tabel 4.11 diperoleh nilai RMSE paling kecil pada sinyal P1 dan paling besar pada sinyal P2. Sedangkan pada Tabel 4.12, nilai RMSE paling kecil didapatkan pada sinyal P2 dan paling besar pada sinyal C1.

Tabel 4. 11 Keakuratan Hasil Koordinat Tanpa Koreksi Troposfer

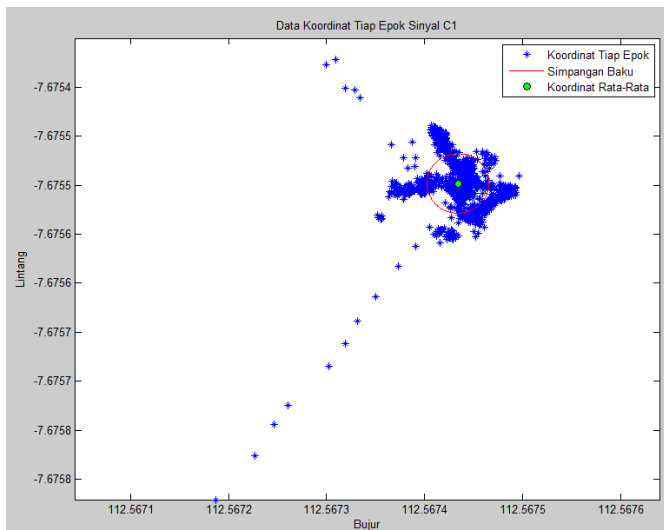
Sinyal	dX (m)	dY (m)	dZ (m)	RMSE (m)
C1	9.678	-19.359	3.594	25.863
P1	9.741	-18.748	3.335	25.357
P2	11.857	-23.257	3.939	30.460

Tabel 4. 12 Keakuratan Hasil Koordinat Dengan Koreksi Troposfer

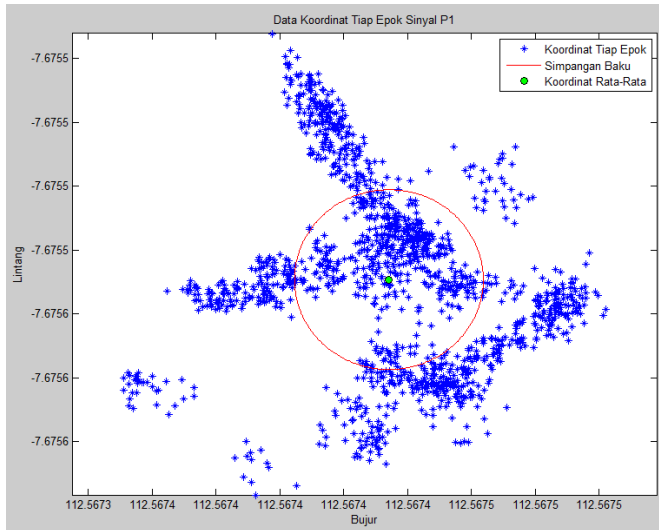
Sinyal	dX (m)	dY (m)	dZ (m)	RMSE (m)
C1	-5.721	16.261	-2.286	20.898
P1	-5.462	15.700	-2.082	19.645
P2	-3.345	11.191	-1.478	16.447

Nilai koordinat titik yang didapatkan pada tabel 4.9 dan 4.10 tersebut merupakan nilai rata – rata dari nilai koordinat titik pada tiap epok pengamatan. Data – data koordinat pada tiap epok tersebut selanjutnya dijumlahkan kemudian dirata – rata sehingga mendapatkan nilai koordinat titik pengamatan.

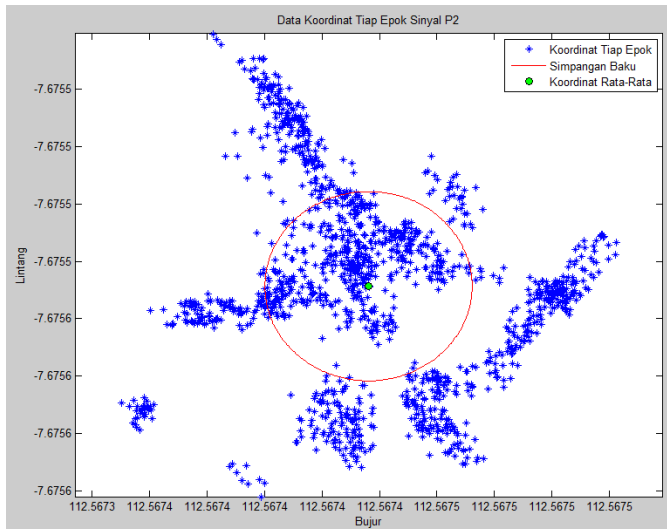
Sebaran nilai koordinat titik pada tiap epok dengan menggunakan koreksi troposfer ditunjukkan pada gambar berikut. Gambar 4.12 menunjukkan sebaran titik koordinat yang dihitung dengan menggunakan data pseudorange pada sinyal C1. Gambar 4.13 menunjukkan sebaran titik koordinat yang dihitung dengan menggunakan data sinyal P1. Gambar 4.14 menunjukkan sebaran titik koordinat yang dihitung dengan menggunakan data sinyal P2. Titik berwarna biru menunjukkan sebaran dari koordinat hasil perhitungan pada setiap epok. Titik berwarna hijau menunjukkan koordinat titik pengamatan hasil rata – rata dari koordinat tiap epok. Dan garis berwarna merah menunjukkan simpangan baku dari hasil perhitungan, dimana titik yang berada di dalam garis merah menunjukkan bahwa titik tersebut dapat diterima.



Gambar 4. 12 Sebaran Koordinat Tiap Epok pada Sinyal C1



Gambar 4. 13 Sebaran Koordinat Tiap Epok pada Sinyal P1



Gambar 4. 14 Sebaran Koordinat Tiap Epok pada Sinyal P2

“Halaman ini sengaja dikosongkan”

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pada penelitian tentang Pembuatan Program Pengolahan Data GPS yang dibuat dengan menggunakan bahasa pemrograman C++ ini dapat disimpulkan adalah sebagai berikut :

1. Program pengolahan data GPS yang telah dibuat dapat dipergunakan untuk pengolahan data RINEX untuk memperoleh koordinat titik pengamatan dengan menggunakan metode penentuan posisi absolut.
2. Hasil perhitungan koordinat dengan menggunakan koreksi troposfer memberikan nilai simpangan baku dan residual terhadap koordinat pendekatan yang lebih kecil daripada perhitungan koordinat tanpa menggunakan koreksi troposfer, karena koreksi troposfer memperkecil nilai bias yang disebabkan oleh lapisan troposfer yang mengakibatkan data pengamatan *pseudorange* lebih panjang.
3. Pada data pengamatan tanggal 23 November 2015 didapatkan nilai simpangan baku terkecil pada sinyal C1 yaitu sebesar 6,085 m dan nilai RMSE terkecil pada P2 yaitu sebesar 12,292 m.
4. Pada data pengamatan tanggal 4 Januari 2016 didapatkan nilai simpangan baku terkecil pada sinyal P1 yaitu sebesar 4,993 m dan nilai RMSE terkecil pada P2 yaitu sebesar 17,828 m.
5. Pada data pengamatan tanggal 5 Januari 2016 didapatkan nilai simpangan baku terkecil pada sinyal P1 yaitu sebesar 5,299 m dan nilai RMSE terkecil pada P2 yaitu sebesar 16,447 m.

6. Data pengamatan tanggal 4 dan 5 Januari 2016 diamati dari titik yang sama dan memberikan hasil nilai koordinat yang hampir sama dengan residual koordinat antara kedua data tersebut adalah sebesar 1,190 m.

5.2 Saran

Saran yang diberikan pada penelitian ini adalah sebagai berikut :

1. Perlu adanya pengembangan lebih lanjut pada program agar metode penentuan posisi yang digunakan lebih beragam.
2. Diharapkan program dapat dikembangkan dengan menggunakan *Graphic User Interface* (GUI) agar lebih mudah digunakan oleh pemakai.
3. Perlu adanya penambahan beberapa macam koreksi pada algoritma perhitungan posisi titik pengamatan agar hasil yang didapatkan dapat lebih akurat dan teliti.

DAFTAR PUSTAKA

- Abidin, H. Z., 2001. *Geodesi Satelit*. Jakarta: Pradnya Paramita.
- Abidin, H. Z., 2007. *Penentuan Posisi Dengan GPS Dan Aplikasinya*. Jakarta: Pradnya Paramita.
- Abidin, H. Z., Jones, A. & Kahar, J., 2002. *Survei Dengan GPS*. Jakarta: Pradnya Paramita.
- Bidikar, B., Rao, G. S., Ganesh, L. & Kumar, M. S., 2014. Satellite Clock Error And Orbital Solution Error Estimation For Precise Navigation Applications. *Scientific Research*, Volume V, pp. 22-26.
- Ghilani, C. D. & Wolf, P. R., 2006. *Adjustment Computation : Spatial Data Analysis*. New Jersey: John Wiley And Sons.
- Kadir, A., 1997. *Pemrograman Dasar TURBO C Untuk IBM PC*. Yogyakarta: Andi Offset.
- Mikhail, E. M. & Gracie, G., 1981. *Analysis And Adjustment Of Survey Measurements*. New York: Van Nostrand Reinhold Company Inc.
- Navstar GPS Joint Program Office, 2004. *IS-GPS-200*. El Segundo: ARINC Engineering Services.
- Rozaq, A., 2012. *Analisa Perbandingan Orbit Satelit GPS Yang Dipengaruhi Oleh Spherically Symmetric Element Keplerian Dan Osculating Element Keplerian*, Surabaya: Teknik Geomatika, Institut Teknologi Sepuluh Nopember.
- Subarya, C., 2004. *Jaring Kontrol Geodesi Nasional Dengan Pengukuran Global Positioning System Dalam ITRF 2000 Epoch 1998.0*. Cibinong: Badan Koordinasi Survei Dan Pemetaan Nasional.

Wells, D. et al., 1999. *Guide To GPS Positioning*. Canada:
University of New Brunswick.

LAMPIRAN

1. Format Data RINEX Navigasi

TABLE A3 GPS NAVIGATION MESSAGE FILE - HEADER SECTION DESCRIPTION			
HEADER LABEL (Columns 61-80)	DESCRIPTION	FORMAT	
RINEX VERSION / TYPE	- Format version (2.10) - File type ('N' for Navigation data)	F9.2,11X, A1,19X	
PGM / RUN BY / DATE	- Name of program creating current file - Name of agency creating current file - Date of file creation	A20, A20, A20	
* COMMENT	Comment line(s)	A60	*
* ION ALPHA	Ionosphere parameters A0-A3 of almanac (page 18 of subframe 4)	2X,4D12.4	*
* ION BETA	Ionosphere parameters B0-B3 of almanac	2X,4D12.4	*
* DELTA-UTC: A0,A1,T,W	Almanac parameters to compute time in UTC (page 18 of subframe 4) A0,A1: terms of polynomial T : reference time for UTC data W : UTC reference week number. Continuous number, not mod(1024)!	3X,2D19.12, 219 *)	*
* LEAP SECONDS	Delta time due to leap seconds	16	*
END OF HEADER	Last record in the header section.	60X	

Records marked with * are optional

TABLE A4 GPS NAVIGATION MESSAGE FILE - DATA RECORD DESCRIPTION			
OBS. RECORD	DESCRIPTION	FORMAT	
PRN / EPOCH / SV CLK	- Satellite PRN number - Epoch: Toc - Time of Clock year (2 digits, padded with 0 if necessary) month day hour minute second - SV clock bias (seconds) - SV clock drift (sec/sec) - SV clock drift rate (sec/sec2)	I2, 1X,I2.2, 1X,I2, 1X,I2, 1X,I2, 1X,I2, F5.1, 3D19.12 *)	

BROADCAST ORBIT - 1	- IODE Issue of Data, Ephemeris - Crcs (meters) - Delta n (radians/sec) - M0 (radians)	3X,4D19.12
BROADCAST ORBIT - 2	- Cus (radians) - e Eccentricity (radians) - Cus (radians) - sqrt(A) (sqrt(m))	3X,4D19.12
BROADCAST ORBIT - 3	- Toe Time of Ephemeris (sec of GPS week) - Cic (radians) - OMEGA (radians) - CIS (radians)	3X,4D19.12
BROADCAST ORBIT - 4	- i0 (radians) - Crc (meters) - omega (radians) - OMEGA DOT (radians/sec)	3X,4D19.12
BROADCAST ORBIT - 5	- IDOT (radians/sec) - Codes on L2 channel - GPS Week # (to go with TOE) - Continuous number, not mod(1024)! - L2 P data flag	3X,4D19.12
BROADCAST ORBIT - 6	- SV accuracy (meters) - SV health (bits 17-22 w 3 sf 1) - TGD (seconds) - IODC Issue of Data, Clock	3X,4D19.12
BROADCAST ORBIT - 7	- Transmission time of message **) (sec of GPS week, derived e.g. from Z-count in Hand Over Word (HOW)) - Fit interval (hours) - (see ICD-GPS-200, 20.3.4.4) - Zero if not known - spare - spare	3X,4D19.12

**) Adjust the Transmission time of message by -604800 to refer to the reported week, if necessary.

*) In order to account for the various compilers, E,e,D, and d are allowed letters between the fraction and exponent of all floating point numbers in the navigation message files.
Zero-padded two-digit exponents are required, however.

2. Format Data RINEX Observasi

TABLE A1 GPS OBSERVATION DATA FILE - HEADER SECTION DESCRIPTION			
HEADER LABEL (Columns 61-80)	DESCRIPTION	FORMAT	
RINEX VERSION / TYPE	- Format version (2.10) - File type ('O' for Observation Data) - Satellite System: blank or 'G': GPS 'R': GLONASS 'S': Geostationary signal payload 'T': NNSS Transit 'M': Mixed	F9.2,11X, A1,19X, A1,19X	
PGM / RUN BY / DATE	- Name of program creating current file - Name of agency creating current file - Date of file creation	A20, A20, A20	
* COMMENT	Comment line(s)	A60	*
MARKER NAME	Name of antenna marker	A60	
* MARKER NUMBER	Number of antenna marker	A20	*
OBSERVER / AGENCY	Name of observer / agency	A20,A40	
REC # / TYPE / VERS	Receiver number, type, and version (Version: e.g. Internal Software Version)	3A20	
ANT # / TYPE	Antenna number and type	2A20	
APPROX POSITION XYZ	Approximate marker position (WGS84)	3F14.4	
ANTENNA: DELTA H/E/N	- Antenna height: Height of bottom surface of antenna above marker - Eccentricities of antenna center relative to marker to the east and north (all units in meters)	3F14.4	
WAVELENGTH FACT L1/2	- Default wavelength factors for L1 and L2 1: Full cycle ambiguities 2: Half cycle ambiguities (squaring) 0 (in L2): Single frequency instrument - zero or blank The default wavelength factor line is required and must precede satellite- specific lines.	2I6, I6	

* WAVELENGTH FACT L1/2	<ul style="list-style-type: none"> - Wavelength factors for L1 and L2 <ul style="list-style-type: none"> 1: Full cycle ambiguities 2: Half cycle ambiguities (squaring) 0 (in L2): Single frequency instrument - Number of satellites to follow in list for which these factors are valid. - List of PRNs (satellite numbers with system identifier) <p>These optional satellite specific lines may follow, if they identify a state different from the default values.</p> <p>Repeat record if necessary.</p>	2I6, I6, 7(3X,A1,I2)	*
# / TYPES OF OBSERV	<ul style="list-style-type: none"> - Number of different observation types stored in the file - Observation types <p>If more than 9 observation types: Use continuation line(s)</p> <p>The following observation types are defined in RINEX Version 2.10:</p> <p>L1, L2: Phase measurements on L1 and L2 C1 : Pseudorange using C/A-Code on L1 P1, P2: Pseudorange using P-Code on L1,L2 D1, D2: Doppler frequency on L1 and L2 T1, T2: Transit Integrated Doppler on 150 (T1) and 400 MHz (T2) S1, S2: Raw signal strengths or SNR values as given by the receiver for the L1,L2 phase observations</p> <p>Observations collected under Antispoofing are converted to "L2" or "P2" and flagged with bit 2 of loss of lock indicator (see Table A2).</p> <p>Units : Phase : full cycles Pseudorange : meters Doppler : Hz Transit : cycles SNR etc : receiver-dependent</p> <p>The sequence of the types in this record has to correspond to the sequence of the observations in the observation records</p>	I6, 9(4X,A2) 6X,9(4X,A2)	
* INTERVAL	Observation interval in seconds	F10.3	*
TIME OF FIRST OBS	<ul style="list-style-type: none"> - Time of first observation record (4-digit-year, month,day,hour,min,sec) - Time system: GPS (=GPS time system) GLO (=UTC time system) Compulsory in mixed GPS/GLONASS files Defaults: GPS for pure GPS files GLO for pure GLONASS files 	5I6,F13.7, 5X,A3	
* TIME OF LAST OBS	<ul style="list-style-type: none"> - Time of last observation record (4-digit-year, month,day,hour,min,sec) - Time system: Same value as in TIME OF FIRST OBS record 	5I6,F13.7, 5X,A3	*

* RCV CLOCK OFFS APPL	Epoch, code, and phase are corrected by applying the realtime-derived receiver clock offset: 1=yes, 0=no; default: 0=no Record required if clock offsets are reported in the EPOCH/SAT records	I6	*
* LEAP SECONDS	Number of leap seconds since 6-Jan-1980 Recommended for mixed GPS/GLONASS files	I6	*
* # OF SATELLITES	Number of satellites, for which observations are stored in the file	I6	*
* PRN / # OF OBS	PRN (sat.number), number of observations for each observation type indicated in the "# / TYPES OF OBSERV" - record. If more than 9 observation types: Use continuation line(s) This record is (these records are) repeated for each satellite present in the data file	3X,A1,I2,9I6 6X,9I6	*
END OF HEADER	Last record in the header section.	60X	
TABLE A2 GPS OBSERVATION DATA FILE - DATA RECORD DESCRIPTION			
OBS. RECORD	DESCRIPTION	FORMAT	
EPOCH/SAT or EVENT FLAG	<ul style="list-style-type: none"> - Epoch : <ul style="list-style-type: none"> - year (2 digits, padded with 0 if necessary) - month,day,hour,min, - sec - Epoch flag 0: OK <ul style="list-style-type: none"> 1: power failure between previous and current epoch >1: Event flag - Number of satellites in current epoch - List of PRNs (sat.numbers with system identifier, see 5.1) in current epoch - receiver clock offset (seconds, optional) If more than 12 satellites: Use continuation line(s) If epoch flag 2-5: <ul style="list-style-type: none"> - Event flag: <ul style="list-style-type: none"> 2: start moving antenna 3: new site occupation (end of kinem. data) (at least MARKER NAME record follows) 4: header information follows 5: external event (epoch is significant, same time frame as observation time tags) - "Number of satellites" contains number of special records to follow. Maximum number of records: 999 - For events without significant epoch the epoch fields can be left blank 	1X,I2.2, 4(1X,I2), F11.7, 2X,I1, I3, 12(A1,I2), F12.9 32X, 12(A1,I2) [2X,I1,] [I3]	

	If epoch flag = 6: 6: cycle slip records follow to optionally report detected and repaired cycle slips (same format as OBSERVATIONS records; slip instead of observation; LLI and signal strength blank or zero)	
OBSERVATIONS	- Observation rep. within record for - LLI each obs.type (same seq - Signal strength as given in header)	m(F14.3, I1, I1)
	If more than 5 observation types (=80 char): continue observations in next record. This record is (these records are) repeated for each satellite given in EPOCH/SAT - record. Observations: Phase : Units in whole cycles of carrier Code : Units in meters Missing observations are written as 0.0 or blanks. Phase values overflowing the fixed format F14.3 have to be clipped into the valid interval (e.g. add or subtract 10**9), set LLI indicator. Loss of lock indicator (LLI). Range: 0-7 0 or blank: OK or not known Bit 0 set : Lost lock between previous and current observation: cycle slip possible Bit 1 set : Opposite wavelength factor to the one defined for the satellite by a previous WAVELENGTH FACT L1/2 line. Valid for the current epoch only. Bit 2 set : Observation under Antispoofing (may suffer from increased noise) Bits 0 and 1 for phase only. Signal strength projected into interval 1-9: 1: minimum possible signal strength 5: threshold for good S/N ratio 9: maximum possible signal strength 0 or blank: not known, don't care	

3. Script Program

Program Utama

```
#include <iostream>
#include <fstream>
#include <math.h>
#include <stdio.h>
#include <conio.h>
#include <iomanip.h>
#include <stdlib.h>
#include <string>
#include <alloc.h>
#include <process.h>
#include "gpsproc.h"
#include "matriks.h"
using namespace std;
int main()
{
    //deklarasi variabel untuk data navigasi
    ifstream navfile;
    int nav = 1000;
    double r_ver, IA[4], IB[4], A0, A1, T, W, leap;
    double PRN[nav], af0[nav], af1[nav], af2[nav];
    double IODE[nav], Crs[nav], Delta_n[nav], M0[nav];
    double Cuc[nav], Ecc[nav], Cus[nav], sqrtA[nav];
    double Toe[nav], Cic[nav], OMEGA0[nav], Cis[nav];
    double i0[nav], Crc[nav], omega[nav], OMEGA_DOT[nav];
    double iDOT[nav], Codes[nav], Week[nav], L2_flag[nav];
    double SV_acc[nav], SV_health[nav], TGD[nav], IODC[nav];
    double Trans[nav];

    //deklarasi variabel untuk data observasi
    ifstream obsfile;
```

```

    int obs = 1000, GPS = 0, GLO = 0, eGPS, eGLO, data_epoch,
    jumlah_data = 0, skip;
    int n, m, n_obs, PRN_numb_GPS[50], PRN_numb_GLO[50],
    pair[50], dim;
    double r_ver2, X0, Y0, Z0, dH, dE, dN;
    std::string t_obs[50], PRN_name;
    std::string PRN_name_epoch;
    double year, month, day, hour, minute, second, He, JD, Tep;
    double flag, n_sat_epoch, PRN_numb_epGPS[50],
    PRN_numb_epGLO[50];
    double L1, L2, C1[50], P1, P2, D1, D2, T1, T2, S1, S2, PS;
    double Tk, Xk[50], Yk[50], Zk[50], Dt[50], El[50], Ds[50],
    Tcor[50], tropo[50], J[50][4], K[50], Lat, Long, Height, Lati,
    Longi, Heighti, Latp, Longp, Heightp;
    double JTK[5][5], JTK[4], **normal, **inv, JX,
    V[50], pos_cor[4], X1, Y1, Z1, Xtot=0, Ytot=0, Ztot=0, Xp, Yp,
    Zp, Sn, S0, Es, Sh, Up, Esp, Shp, Upp, Estot=0, Shtot=0, Uptot=0;

```

```

//deklarasi variabel untuk menyimpan hasil
ofstream ocoord;
char *epoch_coord = "Koordinat Epok.txt";
ocoord.open(epoch_coord);
ofstream ohasil;
char *hasil_akhir = "Koordinat Hasil.txt";
ohasil.open(hasil_akhir);

```

```

//membuka data navigasi
char *rinexnav = "gps0004e.16n";
char *rinexobs = "gps0004e.16o";
int pilih_PR = 32;
navfile.open(rinexnav, ios::in);

```

```

//variabel pengenalan header

```

```

std::string headernav01="RINEX VERSION / TYPE";
std::string headernav02="ION ALPHA      ";
std::string headernav03="ION BETA      ";
std::string headernav04="DELTA-UTC: A0,A1,T,W";
std::string headernav05="LEAP SECONDS  ";
std::string headernav06="END OF HEADER  ";
cout          <<          endl          <<
"=====
===== " << endl;
cout << "Membaca File Navigasi" << endl << endl;
cout << "Data rinex navigasi yang dimasukkan = " << rinexnav
<< endl;

```

```

std::string str;
std::string kosong("      ");
std::string file_contents;
int batas = 0, n_nav = 0;
char ambil[21] = "      ";

```

```

while (std::getline(navfile, str))
{
    file_contents += str;
    file_contents.push_back("\n");
    if (batas==0 && str.length()>=60)
    {
        str.copy(ambil,20,60);
        std::string header(ambil);
        kosong.copy(ambil,20,0);
        if (header.compare(headernav01)==0)
        { str.copy(ambil,9,0);
          r_ver = std::strtod(ambil,0); kosong.copy(ambil,20,0); }
        else if (header.compare(headernav02)==0)
        { str.copy(ambil,12,2);

```

```

IA[0] = std::strtod(ambil,0); kosong.copy(ambil,20,0);
str.copy(ambil,12,14);
IA[1] = std::strtod(ambil,0); kosong.copy(ambil,20,0);
str.copy(ambil,12,26);
IA[2] = std::strtod(ambil,0); kosong.copy(ambil,20,0);
str.copy(ambil,12,38);
IA[3] = std::strtod(ambil,0); kosong.copy(ambil,20,0);}
else if (header.compare(headernav03)==0)
{ str.copy(ambil,12,2);
IB[0] = std::strtod(ambil,0); kosong.copy(ambil,20,0);
str.copy(ambil,12,14);
IB[1] = std::strtod(ambil,0); kosong.copy(ambil,20,0);
str.copy(ambil,12,26);
IB[2] = std::strtod(ambil,0); kosong.copy(ambil,20,0);
str.copy(ambil,12,38);
IB[3] = std::strtod(ambil,0); kosong.copy(ambil,20,0);}
else if (header.compare(headernav04)==0)
{ str.copy(ambil,19,3);
A0 = std::strtod(ambil,0); kosong.copy(ambil,20,0);
str.copy(ambil,19,22);
A1 = std::strtod(ambil,0); kosong.copy(ambil,20,0);
str.copy(ambil,9,41);
T = std::strtod(ambil,0); kosong.copy(ambil,20,0);
str.copy(ambil,9,50);
W = std::strtod(ambil,0); kosong.copy(ambil,20,0);}
else if (header.compare(headernav05)==0)
{ str.copy(ambil,6,0);
leap = std::strtod(ambil,0); kosong.copy(ambil,20,0);}
else if (header.compare(headernav06)==0)
{ batas = 1;}
}
else if (batas==1)
{ str.copy(ambil,2,0);

```

```

        PRN[n_nav]                =                std::strtod(ambil,0);
kosong.copy(ambil,20,0);
        str.copy(ambil,19,22);
        af0[n_nav] = std::strtod(ambil,0); kosong.copy(ambil,20,0);
        str.copy(ambil,19,41);
        af1[n_nav] = std::strtod(ambil,0); kosong.copy(ambil,20,0);
        str.copy(ambil,19,60);
        af2[n_nav] = std::strtod(ambil,0); kosong.copy(ambil,20,0);
        batas = 2;
    }
    else if (batas==2)
    {
        str.copy(ambil,19,3);
        IODE[n_nav]                =                std::strtod(ambil,0);
kosong.copy(ambil,20,0);
        str.copy(ambil,19,22);
        Crs[n_nav] = std::strtod(ambil,0); kosong.copy(ambil,20,0);
        str.copy(ambil,19,41);
        Delta_n[n_nav]            =                std::strtod(ambil,0);
kosong.copy(ambil,20,0);
        str.copy(ambil,19,60);
        M0[n_nav] = std::strtod(ambil,0); kosong.copy(ambil,20,0);
        batas = 3;
    }
    else if (batas==3)
    {
        str.copy(ambil,19,3);
        Cuc[n_nav]                =                std::strtod(ambil,0);
kosong.copy(ambil,20,0);
        str.copy(ambil,19,22);
        Ecc[n_nav] = std::strtod(ambil,0); kosong.copy(ambil,20,0);
        str.copy(ambil,19,41);
        Cus[n_nav]                =                std::strtod(ambil,0);
kosong.copy(ambil,20,0);
        str.copy(ambil,19,60);

```

```

        sqrtA[n_nav]                =                std::strtod(ambil,0);
kosong.copy(ambil,20,0);
        batas = 4;
    }
    else if (batas==4)
    { str.copy(ambil,19,3);
      Toe[n_nav]                    =                std::strtod(ambil,0);
kosong.copy(ambil,20,0);
      str.copy(ambil,19,22);
      Cic[n_nav] = std::strtod(ambil,0); kosong.copy(ambil,20,0);
      str.copy(ambil,19,41);
      OMEGA0[n_nav]                =                std::strtod(ambil,0);
kosong.copy(ambil,20,0);
      str.copy(ambil,19,60);
      Cis[n_nav] = std::strtod(ambil,0); kosong.copy(ambil,20,0);
      batas = 5;
    }
    else if (batas==5)
    { str.copy(ambil,19,3);
      i0[n_nav] = std::strtod(ambil,0); kosong.copy(ambil,20,0);
      str.copy(ambil,19,22);
      Crc[n_nav] = std::strtod(ambil,0); kosong.copy(ambil,20,0);
      str.copy(ambil,19,41);
      omega[n_nav]                =                std::strtod(ambil,0);
kosong.copy(ambil,20,0);
      str.copy(ambil,19,60);
      OMEGA_DOT[n_nav]            =                std::strtod(ambil,0);
kosong.copy(ambil,20,0);
      batas = 6;
    }
    else if (batas==6)
    { str.copy(ambil,19,3);

```

iDOT[n_nav]	=	std::strtod(ambil,0);
kosong.copy(ambil,20,0);		
str.copy(ambil,19,22);		
Codes[n_nav]	=	std::strtod(ambil,0);
kosong.copy(ambil,20,0);		
str.copy(ambil,19,41);		
Week[n_nav]	=	std::strtod(ambil,0);
kosong.copy(ambil,20,0);		
str.copy(ambil,19,60);		
L2_flag[n_nav]	=	std::strtod(ambil,0);
kosong.copy(ambil,20,0);		
batas = 7;		
}		
else if (batas==7)		
{ str.copy(ambil,19,3);		
SV_acc[n_nav]	=	std::strtod(ambil,0);
kosong.copy(ambil,20,0);		
str.copy(ambil,19,22);		
SV_health[n_nav]	=	std::strtod(ambil,0);
kosong.copy(ambil,20,0);		
str.copy(ambil,19,41);		
TGD[n_nav]	=	std::strtod(ambil,0);
kosong.copy(ambil,20,0);		
str.copy(ambil,19,60);		
IODC[n_nav]	=	std::strtod(ambil,0);
kosong.copy(ambil,20,0);		
batas = 8;		
}		
else if (batas==8)		
{ str.copy(ambil,19,3);		
Trans[n_nav]	=	std::strtod(ambil,0);
kosong.copy(ambil,20,0);		
n_nav=n_nav+1;		


```

        batas = 1;
    }
}
cout << "Jumlah data navigasi yang terbaca = " << n_nav <<
endl;
cout << endl << "Selesai membaca data navigasi" << endl;
cout << endl << "=====
===== " << endl;

//membuka data observasi
obsfile.open(rinexobs,ios::in);
//variabel pengenalan header
std::string headerobs01="RINEX VERSION / TYPE";
std::string headerobs02="APPROX POSITION XYZ ";
std::string headerobs03="ANTENNA: DELTA H/E/N";
std::string headerobs04="# / TYPES OF OBSERV ";
std::string headerobs05="INTERVAL ";
std::string headerobs06="TIME OF FIRST OBS ";
std::string headerobs07="TIME OF LAST OBS ";
std::string headerobs08="LEAP SECONDS ";
std::string headerobs09="# OF SATELLITES ";
std::string headerobs10="PRN / # OF OBS ";
std::string headerobs11="END OF HEADER ";
std::string headerobs12="C1 ";
cout << headerobs01 << headerobs02 << headerobs03 <<
headerobs04 << headerobs05 << headerobs06 << headerobs07 <<
headerobs08 << headerobs09 << headerobs10 << headerobs11 <<
headerobs12 << endl << endl;
cout << "Membaca File Observasi" << endl << endl;
cout << "Data rinex observasi yang dimasukkan = " << rinexobs
<< endl;

batas = 0;

```

```

int i = 0, i1, i2, i3, i4, j;

while (std::getline(obsfile, str))
{
    file_contents += str;
    file_contents.push_back("\n");

    if (batas==0 && str.length()>=60)
    {
        str.copy(ambil,20,60);
        std::string header(ambil);
        kosong.copy(ambil,20,0);
        //Membaca header "RINEX VERSION / TYPE"
        if (header.compare(headerobs01)==0)
        {str.copy(ambil,9,0);
         r_ver2 = std::strtod(ambil,0); kosong.copy(ambil,20,0);}
        //Membaca header "APPROX POSITION XYZ "
        else if (header.compare(headerobs02)==0)
        {str.copy(ambil,14,0);
         X0 = std::strtod(ambil,0); kosong.copy(ambil,20,0);
         str.copy(ambil,14,14);
         Y0 = std::strtod(ambil,0); kosong.copy(ambil,20,0);
         str.copy(ambil,14,28);
         Z0 = std::strtod(ambil,0); kosong.copy(ambil,20,0);}
        //Membaca header "ANTENNA: DELTA H/E/N"
        else if (header.compare(headerobs03)==0)
        {str.copy(ambil,14,0);
         dH = std::strtod(ambil,0); kosong.copy(ambil,20,0);
         str.copy(ambil,14,14);
         dE = std::strtod(ambil,0); kosong.copy(ambil,20,0);
         str.copy(ambil,14,28);
         dN = std::strtod(ambil,0); kosong.copy(ambil,20,0);}
        //Membaca header "# / TYPES OF OBSERV "
    }
}

```

```

else if (header.compare(headerobs04)==0)
{str.copy(ambil,6,0);
n_obs = std::strtod(ambil,0); kosong.copy(ambil,20,0);
n=0;
for(n=0; n<n_obs; n++)
{str.copy(ambil,2,(6+4*(n+1)+2*n));
t_obs[n] = ambil; kosong.copy(ambil,20,0); }
//Membaca header "PRN / # OF OBS    "
else if (header.compare(headerobs10)==0)
{str.copy(ambil,1,3);
PRN_name = ambil[0]; kosong.copy(ambil,20,0);
str.copy(ambil,2,4);
if (PRN_name == "G")
{PRN_numb_GPS[GPS]      =      std::strtod(ambil,0);
kosong.copy(ambil,20,0); GPS++;}
else
{PRN_numb_GLO[GLO]      =      std::strtod(ambil,0);
kosong.copy(ambil,20,0); GLO++;}}
else if (header.compare(headerobs11)==0)
{batas = 1;}
}
else if (batas==1)
{ i1 = 0; i2 = 0; skip = 0;
for (i1 = 0; i1 < 4; i1++)
{JTK[i1] = 0; pos_cor[i1] = 0;
for (i2 = 0; i2 < 4; i2++)
{JTJ[i1][i2] = 0;}}
data_epoch = 0; Sn = 0;
eGPS = 0; eGLO = 0;
str.copy(ambil,2,1);
year      =      2000      +      std::strtod(ambil,0);
kosong.copy(ambil,20,0);
str.copy(ambil,2,4);

```

```

    month = std::strtod(ambil,0); kosong.copy(ambil,20,0);
    str.copy(ambil,2,7);
    day = std::strtod(ambil,0); kosong.copy(ambil,20,0);
    str.copy(ambil,2,10);
    hour = std::strtod(ambil,0); kosong.copy(ambil,20,0);
    str.copy(ambil,2,13);
    minute = std::strtod(ambil,0); kosong.copy(ambil,20,0);
    str.copy(ambil,11,15);
    second = std::strtod(ambil,0); kosong.copy(ambil,20,0);
    str.copy(ambil,1,28);
    flag = std::strtod(ambil,0); kosong.copy(ambil,20,0);
    if (flag != 0)
    { cout << "Data error pada epoch ke - " << jumlah_data+1
    << endl;
    break;}
    str.copy(ambil,3,29);
    //Perhitungan Data Epoch
    He = hour + minute/60 + second/3600;
    if (month<=2)
    { month = month + 12; year = year - 1;}
    JD
    =
    floor(365.25*(year+4716))+floor(30.6001*(month+1))+day+He/2
    4-1537.5;
    Tep = sec_week(JD);
    //membaca nomor satelit
    n_sat_epoch = std::strtod(ambil,0);
    kosong.copy(ambil,20,0);
    n=0;
    for (n=0; n<12; n++)
    { str.copy(ambil,1,32+n*3);
    PRN_name_epoch = ambil[0]; kosong.copy(ambil,20,0);
    str.copy(ambil,2,33+n*3);
    if (PRN_name_epoch == "G")

```

```

        {PRN_numb_epGPS[eGPS]      =      std::strtod(ambil,0);
kosong.copy(ambil,20,0);
        m = 0;
        for (m=0; m<n_nav; m++)
        {if (PRN_numb_epGPS[eGPS]==PRN[m])
            {if (Tep - Toe[m] >= -7200)
                {pair[eGPS] = m;}} }
        eGPS++;}
        else
        {PRN_numb_epGLO[eGLO]      =      std::strtod(ambil,0);
kosong.copy(ambil,20,0); eGLO++;}}

        if (n_sat_epoch > 12)
        {batas = 2;}
        else
        {batas = 3;}
        jumlah_data++;
    }
    else if (batas==2)
    {
        if (flag != 0) break;
        n=12;
        for (n=12; n<n_sat_epoch; n++)
        {str.copy(ambil,1,32+(n-12)*3);
        PRN_name_epoch = ambil[0]; kosong.copy(ambil,20,0);
        str.copy(ambil,2,33+(n-12)*3);
        if (PRN_name_epoch == "G")
        {PRN_numb_epGPS[eGPS]      =      std::strtod(ambil,0);
kosong.copy(ambil,20,0); eGPS++;}
        else
        {PRN_numb_epGLO[eGLO]      =      std::strtod(ambil,0);
kosong.copy(ambil,20,0); eGLO++;}}
        batas = 3;

```

```

    }
else if (batas==3)
{
    if (flag != 0) break;
    str.copy(ambil,16,pilih_PR);
    PS = std::strtod(ambil,0); kosong.copy(ambil,20,0);
    data_epoch++;
    if (PS == 0)
    { skip++; }
    else {
        if (data_epoch <= eGPS)
        {
            m = data_epoch-1-skip;
            C1[m] = PS;

            n = pair[m+skip];
            Xk[m] = sat_pos(Tep, C1[m], Toe[n], af0[n], af1[n], af2[n],
sqrtA[n], Delta_n[n], M0[n], Ecc[n], omega[n], Cuc[n], Cus[n],
Crc[n], Crs[n], Cic[n], Cis[n], i0[n], iDOT[n], OMEGA0[n],
OMEGA_DOT[n], X0, Y0, Z0, 1);
            Yk[m] = sat_pos(Tep, C1[m], Toe[n], af0[n], af1[n], af2[n],
sqrtA[n], Delta_n[n], M0[n], Ecc[n], omega[n], Cuc[n], Cus[n],
Crc[n], Crs[n], Cic[n], Cis[n], i0[n], iDOT[n], OMEGA0[n],
OMEGA_DOT[n], X0, Y0, Z0, 2);
            Zk[m] = sat_pos(Tep, C1[m], Toe[n], af0[n], af1[n], af2[n],
sqrtA[n], Delta_n[n], M0[n], Ecc[n], omega[n], Cuc[n], Cus[n],
Crc[n], Crs[n], Cic[n], Cis[n], i0[n], iDOT[n], OMEGA0[n],
OMEGA_DOT[n], X0, Y0, Z0, 3);
            Tcor[m] = sat_pos(Tep, C1[m], Toe[n], af0[n], af1[n],
af2[n], sqrtA[n], Delta_n[n], M0[n], Ecc[n], omega[n], Cuc[n],
Cus[n], Crc[n], Crs[n], Cic[n], Cis[n], i0[n], iDOT[n],
OMEGA0[n], OMEGA_DOT[n], X0, Y0, Z0, 4);

```

```

    Dt[m] = sat_pos(Tep, C1[m], Toe[n], af0[n], af1[n], af2[n],
sqrtA[n], Delta_n[n], M0[n], Ecc[n], omega[n], Cuc[n], Cus[n],
Crc[n], Crs[n], Cic[n], Cis[n], i0[n], iDOT[n], OMEGA0[n],
OMEGA_DOT[n], X0, Y0, Z0, 5);
    Ds[m] = toposentrik(X0,Y0,Z0,Xk[m],Yk[m],Zk[m],3);
    El[m] = toposentrik(X0,Y0,Z0,Xk[m],Yk[m],Zk[m],2);
    tropo[m] = tropos(El[m],0,1013,293,50,0,0,0);
    J[m][0] = (X0-Xk[m])/C1[m]; J[m][1] = (Y0-
Yk[m])/C1[m]; J[m][2] = (Z0-Zk[m])/C1[m]; J[m][3] = 1;
    K[m] = C1[m]-Ds[m]+Dt[m]-tropo[m];
}
}
if (data_epoch == (eGPS))
{
    i1 = 0; i2 = 0; i3 = 0, i4 = 0;
    for (i1=0; i1<4; i1++)
    { for (i2=0; i2<4; i2++)
    { for (i3=0; i3<eGPS-skip; i3++)
    { J TJ[i1][i2] = J TJ[i1][i2] + J[i3][i1]*J[i3][i2];
    }
    }
    for (i4=0; i4<eGPS-skip; i4++)
    { J TK[i1] = J TK[i1] + K[i4]*J[i4][i1];
    }
    }
}
if (data_epoch == (eGPS))
{ dim = 4;
    normal = matriks(1,dim,1,dim);
    for (i=1;i<=dim;i++){
    for (j=1;j<=dim;j++){
    normal[i][j] = J TJ[i-1][j-1];}}
    inv = invers(dim, normal);

```

```

    for (i=1;i<=dim;i++){
    for (j=1;j<=dim;j++){
    pos_cor[i-1] = pos_cor[i-1] + JTK[j-1]*inv[i][j];}}
    for (i3=0; i3<eGPS-skip; i3++)
    {JX = 0;
    for (i4=0; i4<dim; i4++)
    {JX = JX + J[i3][i4]*pos_cor[i4];}
    V[i3] = JX - K[i3];
    Sn = Sn + V[i3]*V[i3];}
    }
    if (data_epoch == (eGPS))
    {X1 = X0 + pos_cor[0]; Y1 = Y0 + pos_cor[1]; Z1 = Z0 +
pos_cor[2];
    S0 = sqrt(Sn/(eGPS-skip-4));
    Xtot = Xtot + X1; Ytot = Ytot + Y1; Ztot = Ztot + Z1;
    Lat = toposentrik(X1,Y1,Z1,X0,Y0,Z0,4);
    Long = toposentrik(X1,Y1,Z1,X0,Y0,Z0,5);
    Height = toposentrik(X1,Y1,Z1,X0,Y0,Z0,6);
    Es = toposentrik(X0,Y0,Z0,X1,Y1,Z1,7);
    Sh = toposentrik(X0,Y0,Z0,X1,Y1,Z1,8);
    Up = toposentrik(X0,Y0,Z0,X1,Y1,Z1,9);
    Estot = Estot + Es; Shtot = Shtot + Sh; Uptot = Uptot + Up;
    ocoord << fixed << setprecision(5) << X1 << " " << Y1 <<
" " << Z1 << " " << S0 << " " << Height << " " ;
    ocoord << fixed << setprecision(9) << Lat << " " << Long
<< " " ;
    ocoord << fixed << setprecision(1) << jumlah_data << " "
<< m << endl;}
    batas = 4;
    }
    else if (batas==4)
    {
    if (flag != 0) break;

```


[illegible]

```

    ohasil << "Z = " << fixed << setprecision(5) << Zp << " m" <<
endl;
    ohasil << endl << "Hasil perhitungan koordinat geografis titik
receiver :" << endl;
    ohasil << "Lintang = " << fixed << setprecision(9) << Latp << "
derajat" << endl;
    ohasil << "Bujur = " << fixed << setprecision(9) << Longp << "
derajat" << endl;
    ohasil << "Tinggi = " << fixed << setprecision(5) << Heightp
<< " m" << endl;

```

```

ifstream icoord;
icoord.open(epoch_coord,ios::in);
i = 0;
double Xi, Yi, Zi, dXi, dYi, dZi, SXi = 0, SYi = 0, SZi = 0, devX,
devY, devZ, ni, mi, Si, Ssum=0, Smean, Esi, Shi, Upi, dEsi, dShi,
dUpi, SEsi = 0, SShi = 0, SUpi = 0, devEs, devSh, devUp;
for (i = 0; i < jumlah_data; i++)
{ icoord >> Xi >> Yi >> Zi >> Si >> Heighti >> Lati >> Longi
>> ni >> mi ;
    Ssum = Ssum+Si;
    dXi = pow((Xi-Xp),2);
    SXi = SXi + dXi;
    dYi = pow((Yi-Yp),2);
    SYi = SYi + dYi;
    dZi = pow((Zi-Zp),2);
    SZi = SZi + dZi;
    dEsi = pow((Esi-Esp),2);
    SEsi = SEsi + dEsi;
    dShi = pow((Shi-Shp),2);
    SShi = SShi + dShi;
    dUpi = pow((Upi-Upp),2);
    SUpi = SUpi + dUpi;}

```

```

    Smean = Ssum/jumlah_data;
    devX = sqrt(SXi/(jumlah_data));
    devY = sqrt(SYi/(jumlah_data));
    devZ = sqrt(SZi/(jumlah_data));
    devEs = sqrt(SEsi/(jumlah_data));
    devSh = sqrt(SShi/(jumlah_data));
    devUp = sqrt(SUpi/(jumlah_data));
    cout << endl << "Simpangan baku rata - rata :" << endl;
    printf("S = %5.5f m\n",Smean);
    ohasil << endl << "Simpangan baku koordinat titik receiver :"
<< endl;
    ohasil << "SX = " << fixed << setprecision(5) << devX << " m"
<< endl;
    ohasil << "SY = " << fixed << setprecision(5) << devY << " m"
<< endl;
    ohasil << "SZ = " << fixed << setprecision(5) << devZ << " m"
<< endl;
    ohasil << "SE = " << fixed << setprecision(5) << devEs << " m"
<< endl;
    ohasil << "SN = " << fixed << setprecision(5) << devSh << " m"
<< endl;
    ohasil << "SU = " << fixed << setprecision(5) << devUp << "
m" << endl;
    ohasil << endl << "Simpangan baku rata - rata :" << endl;
    ohasil << "S = " << fixed << setprecision(5) << Smean << " m"
<< endl;
    getche();
}

```

Fungsi Perhitungan

//gpsproc.h fungsi - fungsi untuk perhitungan data pengukuran GPS

```
double sec_week(double julian_date);
```

```

double week(double julian_date);
double sat_pos(double Time, double PsRange, double Toe, double
a0, double a1, double a2, double sqrtA, double Delta_n, double
M0, double Ecc, double omega, double Cuc, double Cus, double
Crc, double Crs, double Cic, double Cis, double i0, double idot,
double OMEGA0, double OMEGA_DOT, double X0, double Y0,
double Z0, double tipe);
double toposentrik(double X0,double Y0, double Z0, double Xsat,
double Ysat, double Zsat, double tipe);
double tropos(double sinel,double hsta,double p,double
tkel,double hum,double hp,double htkel,double hhum);

```

```

double sec_week(double julian_date)
{ double a, b, c, d, e, f, day_of_week, week, sec_of_week;
  a = floor(julian_date+0.5);
  b = a + 1537;
  c = floor((b-122.1)/365.25);
  e = floor(365.25*c);
  f = floor((b-e)/30.6001);
  d = b-e-floor(30.6001*f)+fmod(julian_date+0.5,1);
  day_of_week = fmod(floor(julian_date+0.5),7);
  sec_of_week = (fmod(d,1)+day_of_week+1)*86400;
  return sec_of_week;
}

```

```

double week(double julian_date)
{ double a, b, c, d, e, f, day_of_week, week, sec_of_week;
  week = floor((julian_date-2444244.5)/7);
  return week;
}

```

```

double sat_pos(double Time, double PsRange, double Toe, double
a0, double a1, double a2, double sqrtA, double Delta_n, double

```

```

M0, double Ecc, double omega, double Cuc, double Cus, double
Crc, double Crs, double Cic, double Cis, double i0, double idot,
double OMEGA0, double OMEGA_DOT, double X0, double Y0,
double Z0, double tipe)
{ double v_light = 299792458, GM = 3.986005e14, OMEGAe_dot
= 7.2921151467e-5, pi = 3.14159265359;
    double j, Ts, dt, Tcor, Tgps, A, Tk, n0, n, M, E, E_old, dE, v, phi,
u, r, i, OMEGA, x1, y1, X, Y, Z, rho2, Ttrav, OMEGA_tau, Xr, Yr,
Zr, Dt;
    Ts = Time-PsRange/v_light;
    dt = Ts-Toe;
    Tcor = a0+(a1+a2*dt)*dt;
    Tgps = Ts-Tcor;
    dt = Tgps-Toe;
    Tcor = a0+(a1+a2*dt)*dt;
    Dt = Tcor*v_light;
    Tgps = Ts-Tcor;
    A = sqrtA*sqrtA;
    Tk = Tgps-Toe;
    n0 = sqrt(GM/(A*A*A));
    n = n0+Delta_n;
    M = M0+n*Tk;
    M = fmod(M+2*pi,2*pi);
    E = M;
    j = 0;
    for (j=0; j<10; i++)
    { E_old = E;
      E = M+Ecc*sin(E);
      dE = fmod(E-E_old,2*pi);
      if (fabs(dE) < 1e-12)
        break;}
    E = fmod(E+2*pi,2*pi);
    v = atan2(sqrt(1-Ecc*Ecc)*sin(E), cos(E)-Ecc);

```

```

phi = v+omega;
u = phi          +Cuc*cos(2*phi)+Cus*sin(2*phi);
r = A*(1-Ecc*cos(E))+Crc*cos(2*phi)+Crs*sin(2*phi);
i = i0+idot*Tk   +Cic*cos(2*phi)+Cis*sin(2*phi);
OMEGA  =  OMEGA0+(OMEGA_DOT-OMEGAe_dot)*Tk-
OMEGAe_dot*Toe;
x1 = cos(u)*r;
y1 = sin(u)*r;
X = x1*cos(OMEGA)-y1*cos(i)*sin(OMEGA);
Y = x1*sin(OMEGA)+y1*cos(i)*cos(OMEGA);
Z = y1*sin(i);
rho2 = (X-X0)*(X-X0)+(Y-Y0)*(Y-Y0)+(Z-Z0)*(Z-Z0);
Ttrav = sqrt(rho2)/v_light;
OMEGA_tau = OMEGAe_dot*Ttrav;
Xr = X*cos(OMEGA_tau)+Y*sin(OMEGA_tau);
Yr = -X*sin(OMEGA_tau)+Y*cos(OMEGA_tau);
Zr = Z;
rho2 = (Xr-X0)*(Xr-X0)+(Yr-Y0)*(Yr-Y0)+(Zr-Z0)*(Zr-Z0);
if (tipe == 1)
return Xr;
else if (tipe == 2)
return Yr;
else if (tipe == 3)
return Zr;
else if (tipe == 4)
return Tcor;
else if (tipe == 5)
return Dt;
}

```

```

double toposentrik(double X0,double Y0, double Z0, double Xsat,
double Ysat, double Zsat, double tipe)

```

```

{ double pi = 3.14159265359, a = 6378137, f = 298.257223563,
tolsq = 1e-10, i = 0;
  double dX, dY, dZ, dtr, rtd, esq, onesq, P, dlambda, r, sinphi,
dphi, h, cosphi, N_phi, dP, dZe;
  double cl, sl, cb, sb, E, N, U, hor_dis, Az, El, D;
  dX = Xsat-X0;
  dY = Ysat-Y0;
  dZ = Zsat-Z0;
  rtd = 180/pi;
  dtr = pi/180;
  //Perhitungan koordinat ke geodetik
  if (f < 1e-20)
    {esq = 0;}
  else
    {esq = (2-1/f)/f;}
  onesq = 1-esq;
  P = sqrt(X0*X0+Y0*Y0);
  if (P > 1e-20)
    {dlambda = atan2(Y0,X0)*rtd;}
  else
    {dlambda = 0;}
  if (dlambda < 0)
    {dlambda = dlambda+360;}
  r = sqrt(P*P+Z0*Z0);
  if (r < 1e-20)
    {sinphi = Z0/r;}
  else
    {sinphi = 0;}
  dphi = asin(sinphi);
  h = r-a*(1-sinphi*sinphi/f);
  dphi = atan(Z0/(P*(1-esq)));
  h = P/cos(dphi);
  for (i=0; i<10; i++)

```

```

{ sinphi = sin(dphi);
  cosphi = cos(dphi);
  N_phi = a/sqrt(1-esq*sinphi*sinphi);
  dP = P-(N_phi+h)*cosphi;
  dZe = Z0-(N_phi*onesq+h)*sinphi;
  h = h+(sinphi*dZe+cosphi*dP);
  dphi = dphi+(cosphi*dZe-sinphi*dP)/(N_phi+h);
  dphi = atan((Z0+esq*N_phi*sinphi)/P);
  h = P/cos(dphi) - N_phi;
  if ((dP*dP+dZe*dZe) < tolsq)
    break;}
dphi = dphi*rtd;

```

```

cl = cos(dlambda*dtr);
sl = sin(dlambda*dtr);
cb = cos(dphi*dtr);
sb = sin(dphi*dtr);
E = -sl*dX+cl*dY+0*dZ;
N = -sb*cl*dX-sb*sl*dY+cb*dZ;
U = cb*cl*dX+cb*sl*dY+sb*dZ;
hor_dis = sqrt(E*E+N*N);
if (hor_dis < 1e-20)
{ Az = 0;
  El = 90;}
else
{ Az = atan2(E,N)/dtr;
  El = atan2(U,hor_dis)/dtr;}
if (Az < 0)
{ Az = Az+360;}
D = sqrt(dX*dX+dY*dY+dZ*dZ);
if (tipe == 1)
return Az;
else if (tipe == 2)

```



```

return El;
else if (tipe == 3)
return D;
else if (tipe == 4)
return dphi;
else if (tipe == 5)
return dlambd;
else if (tipe == 6)
return h;
else if (tipe == 7)
return E;
else if (tipe == 8)
return N;
else if (tipe == 9)
return U;
}

```

```

double tropos(double el,double hsta,double p,double tkel,double
hum,double hp,double htkel,double hhum)
{ double pi = 3.14159265359, a_e = 6378.137, b0 = 7.839257e-5,
tlapse = -6.5, tropo = 0, done = 0;
double tkhum, atkel, e0, tksea, em, tkelh, e0sea, tkelp, psea,
refsea, htop, ref, rtop, a, b, rn[8], alpha[8], dr, ddr, dtr, sinel;
dtr = pi/180;
sinel = sin(el*dtr);
tkhum = tkel+tlapse*(hhum-htkel);
atkel = 7.5*(tkhum-273.15)/(237.3+tkhum-273.15);
e0 = 0.0611*hum*pow(10,atkel);
tksea = tkel-tlapse*htkel;
em = -978.77/(2.8704e6*tlapse*1e-5);
tkelh = tksea+tlapse*hhum;
e0sea = e0*pow((tksea/tkelh),(4*em));
tkelp = tksea+tlapse*hp;

```

```

psea = p*pow((tksea/tkelp),em);
if (sinel < 0)
    { sinel = 0;}
refsea = 77.624e-6/tksea;
htop = 1.1385e-5/refsea;
refsea = refsea*psea;
ref = refsea*pow(((htop-hsta)/htop),4);
while (1)
{ rtop = pow((a_e+htop),2)-pow((a_e+hsta),2)*(1-pow(sinel,2));
if (rtop < 0)
    { rtop = 0;}
rtop = sqrt(rtop)-(a_e+hsta)*sinel;
a = -sinel/(htop-hsta);
b = -b0*(1-pow(sinel,2))/(htop-hsta);
int i = 0;
for (i = 0; i<8; i++)
    { rn[i] = pow(rtop,(i+1));}
alpha[0] = 2*a;
alpha[1] = 2*pow(a,2)+4*b/3;
alpha[2] = a*(pow(a,2)+3*b);
alpha[3] = pow(a,4)/5+2.4*pow(a,2)*b+1.2*pow(b,2);
alpha[4] = 2*a*b*(pow(a,2)+3*b)/3;
alpha[5] = pow(b,2)*(6*pow(a,2)+4*b)*1.428571e-1;
alpha[6] = 0;
alpha[7] = 0;
if (pow(b,2) > 1.0e-35)
    { alpha[6] = a*pow(b,3)/2;
      alpha[7] = pow(b,4)/9;}
dr = rtop;
i=0;
for (i = 0; i<8; i++)
    { dr = dr+alpha[i]*rn[i];}
tropo = tropo+dr*ref*1000;

```

```
if (done == 1)
{ ddr = tropo;
  break;}
done = 1;
refsea = (371900.0e-6/tksea-12.92e-6)/tksea;
htop = 1.1385e-5*(1255/tksea+0.05)/refsea;
ref = refsea*e0sea*pow(((htop-hsta)/htop),4);}
return tropo;
}
```

BIODATA PENULIS



Penulis dilahirkan di Jember, pada tanggal 21 September 1994, merupakan anak pertama dari dua bersaudara. Penulis telah menempuh pendidikan formal di TK Bhayangkara, SDN 1 Genteng, SMPN 1 Genteng, SMAN 1 Genteng. Setelah lulus dari SMA memilih untuk melanjutkan kuliah S-1 dengan mengikut program SNMPTN dan diterima di Teknik Geomatika – FTPS ITS pada tahun 2013, terdaftar dengan NRP 3513100047. Selama menjadi mahasiswa Teknik Geomatika, penulis aktif dalam kegiatan organisasi dan keilmiahan. Pada tahun 2014 penulis aktif menjadi anggota organisasi pada tingkat fakultas yaitu BEM FTSP sebagai anggota departemen Riset dan Teknologi dan pada tingkat jurusan yaitu HIMAGE sebagai anggota departemen Keilmiahan dan Keprofesian. Pada tahun 2015 penulis aktif dalam organisasi HIMAGE sebagai kepala departemen Keilmiahan dan Keprofesian. Penulis juga aktif mengikuti kegiatan yang bersifat keilmiahan seperti Program Kreativitas Mahasiswa (PKM). Penulis melaksanakan kegiatan Kerja Praktik di Pusat Pemanfaatan Penginderaan Jauh Lembaga Antariksa Dan Penerbangan Nasional (LAPAN) untuk mengidentifikasi fase pertumbuhan tanaman padi dengan menggunakan data citra satelit Landsat-8 multitemporal. Penulis menyelesaikan studi program sarjana di Teknik Geomatika ITS dengan mengambil Tugas Akhir bidang keahlian Geodinamika dan Lingkungan, yang berjudul **“Pengembangan Program Pengolahan Data GPS : Global Positioning Software”**.

“Halaman ini sengaja dikosongkan”